NO. 14-1401

IN THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

SAMSUNG ELECTRONICS CORPORATION, LTD.,

Appellant

v.

CCP SYSTEMS AG,

Cross-Appellant

BRIEF OF APPELLANT SAMSUNG ELECTRONICS CORPORATION, LTD.

From the U.S. Patent and Trademark Office, PTAB Appeal No. 2013-009045 , Hon. Howard B. Blankenship, Stanley M. Weinberg, and Stacey G. White, Administrative Patent Judges

David L. McCombs
Debra J. McComas
HAYNES AND BOONE LLP
2323 Victory Avenue, Suite 700
Dallas, Texas 75219
Telephone: 214.651.5375
Facsimile: 214.200.0525
David.McCombs@haynesboone.com
Debbie.McComas@haynesboone.com

Attorneys For Appellant,
Samsung Electronics Corporation, Ltd.

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rules 21(a)(2) and 47.4(a)(1), counsel for the Appellant Samsung Electronics Corporation, Ltd. certifies the following:

1. The full name of every party or amicus represented by me is:

Samsung Electronics Corporation, Ltd.

2. The name of the real party in interest (If the party named in the caption is not the real party in interest) represented by me is:

None.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

There is no parent corporation or any publicly held company that owns 10 percent or more of Samsung Electronics Corporation, Ltd.

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

Haynes and Boone, LLP
David L. McCombs
Debra J. McComas

Dated this 20th day of June, 2014.

/s/ David L. McCombs
David L. McCombs

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

STATEMENT OF RELATED CASES

This appeal arises from a final decision of the Patent Trial and Appeal Board on *inter partes* reexamination rejecting all but two claims of United States Patent No. 6,684,789 (the '789 Patent). No "appeal in or from the same [*inter partes* reexamination] was previously before this or any other appellate court." Fed. Cir. R. 47.5. However, the '789 Patent remains the subject of pending litigation in the United States District Court for the District of New Jersey, in *CCP Systems AG v. Samsung Electronics Corp.*, No. 2:09-cv-04354-DMC-CCC.

JURISDICTIONAL STATEMENT

This Court has jurisdiction over this timely appeal from the final Decision on Appeal of the Patent Trial and Appeal Board in an *inter partes* reexamination proceeding pursuant to 28 U.S.C. § 1295(a)(4)(A).

STATEMENT OF THE ISSUES

This appeal arises from an *inter partes* reexamination of the '789 Patent. The Examiner rejected every claim of the '789 Patent. On appeal to the Board, all of the rejections were affirmed, save two new dependent claims (55 and 63). The Appellant Samsung Electronics Corporation, Ltd. ("Samsung") challenges the reversal of the rejection of these two remaining claims 55 and 63.[1] This narrow appeal raises the following issues:

1. Did the Board err in reversing the rejection of claims 55 and 63 under 35 U.S.C. § 103(a) as obvious?

2. Did the Board err in reversing the rejection of claim 63 under 35 U.S.C. § 112 as indefinite?

[1] The Examiner rejected claims 55, 63, 67, and 74 under 35 U.S.C. § 103 as obvious in light of various combinations of prior art with the '705 Patent. (A0006, 2426-27.) The Board reversed the rejection on this ground but affirmed the rejection of claims 67 and 74 on other grounds. (A0021.) The Examiner further rejected claims 59-64, 66, and 67 as indefinite. (A0021.) The Board reversed on this ground but affirmed the rejection of claims 59-62, 64, 66, and 67 on other grounds. (A0021.) Because claims 55, 59-64, 66-67 and 74 stand rejected, Samsung limits its appeal to claims 55 and 63, while reserving its right to assert the same grounds of obviousness and indefiniteness as to the rejected claims in response to any appeal by CCP. *See, e.g., Rexnord Indus., Inc. v. Kappos*, 705 F.3d 1347, 1356 (Fed. Cir. 2013) ("on judicial review, the correctness of the decision appealed from can be defended by the appellee on any ground that is supported by the record, whether or not the appellant raised the argument.") (emphasis in original) (citing *Jaffke v. Dunham*, 352 U.S. 280, 281 (1957)).

STATEMENT OF THE CASE

This appeal arises from an *inter partes* reexamination of the '789 Patent, which relates to a system and method for transforming digital print data streams sent to an intelligent output device such as a printer. (A0046, 0069 at 8:55-61, 0065 at Fig. 1.) Samsung filed a request for *inter partes* reexamination on July 16, 2010 asserting that all claims of the '789 Patent—claims 1-53—are invalid over prior art references that were not previously before the Patent Office. (A0044.) The Examiner granted Samsung's request and issued a first office action on the merits, rejecting all 53 claims. (A2115.)

The patent owner CCP Systems AG (CCP) responded and sought to amend claim 9 of the '789 Patent and to add claims 54-82. (A1074-1220.) After briefing by both parties, the Examiner issued an Action Closing Prosecution (ACP) and maintained the rejection of claims 1-53 and further rejected the newly-added claims 54-82. (A2328.) The Examiner further maintained all of the rejections in the Right of Appeal Notice (RAN). (A2393-2442.)

On appeal, the Board affirmed the rejection of all claims of the '789 Patent except claims 55 and 63. (A0001-0023.) The parties both filed timely notices of appeal. (A2613-2616.) Samsung challenges only the failure to reject claims 55 and 63 of the '789 Patent.

STATEMENT OF FACTS

This case comes to the Court on cross-appeals from the Board's Decision rejecting all but two of the claims of the '789 Patent. Samsung's appeal is limited to the Board's Decision reversing the rejection of those two remaining claims—claims 55 and 63. The facts specific to the rejection of claims 55 and 63 are found in parts A and B.5 of the Statement of Facts. However, to give the Court context for the arguments, including CCP's cross-appeal, Samsung provides a more complete factual statement below.

A. The '789 Patent relates to printer technology.

The '789 Patent relates to printer technology and specifically, a system and method for transforming digital print data streams sent to an intelligent output device as a printer. (A0069 at 8:55-61; A0065 at Fig. 1.)

The '789 Patent initially contained 53 claims. (A0070-72.) Each of the independent claims describes

- receiving and "parsing"
- digital print data streams
- into "graphically representable objects"
- stored in an "object oriented format."

(*See, e.g.,* A0070 at 9:30-35.) The specification of the '789 Patent describes, *inter alia*, lines, rectangles, polygons, and bar codes as examples of graphically

represented objects. (A0067 at 3:38-52.) Each independent claim further describes assigning a "script" to at least one of the objects. (*See, e.g.,* A0070 at 9:41-43.) The specification describes several examples of scripts, including a script that instructs the printer to fold the printer output in a certain way, and a script that instructs the printer to access a database interactively during the printing operation. (A0068 at 5:3-10; 6:47-51.)

The application for the '789 Patent was allowed in 2004 after the Examiner found that "the prior art fails to teach the entire combination of a method for the transformation of digital print data stream including a [*sic*] steps of" the following:

- Reading an input print data stream
- Analyzing the input data stream by means of a parser
- Splitting the input data into the graphically representable objects
- Storing the graphically representable objects in a memory in an object-oriented format
- Transforming the object-oriented format into a format for controlling a printer
- The object-oriented format stored in the memory including at least one stored script.

(A0046; A1008; A0180.) But in reaching this conclusion, the Examiner did not have before it key references that teach these very features.

B. Samsung files a request for *inter partes* reexamination of the '789 Patent.

On July 16, 2010, Samsung requested an *inter partes* reexamination of all 53 claims of the '789 Patent based on five prior art references that were not before the Patent Office in the original office action and, individually, or in combination, raised substantial new questions of patentability. Those new prior art references included the following:

- *IBM* – AS/400 Guide to Advanced Function Presentation and Print Services Facility, Fourth Edition, IBM Printing Systems, January 1999.
- *Interleaf* – English, Paul M. et al., Interleaf active documents, Electronic Publishing, Vol. 7(2), pp. 75-87, June 1994.
- *Interleaf Patent* – U.S. Patent No. 5,579,519 to Pelletier, filed February 16, 1994, issued November 26, 1996.
- *Lieberman* – Lieberman, Henry, "Integrating user interface agents with conventional applications," Knowledge-Based Systems 11 (1998), pp. 15-23.
- *'705 Patent* – U.S. Patent No. 6,678,705 to Barchtold et al., filed November 16, 1999, issued January 13, 2004.

(A0047.) It is undisputed that these prior art references were not before the Patent Office in the initial office action and are not cumulative of art before the Patent Office. Each of these prior art references is summarized in more detail below.

1. IBM describes the features listed in the Examiner's reasons for allowance.

The IBM reference was published on January 1999, and qualifies as prior art under 35 U.S.C. § 102(b). IBM is directed to presentation and printer operations of

IBM's midrange computer, the AS/400. (A0053.) These operations are performed in a subsystem called the "Advanced Function Presentation" and printing system, also called the "AFP." (A0050; A0225.) IBM describes the AFP Data Stream as including both data and text and being independent of operating systems and "portable across environments." (A0053; A0239.)

AFP utilizes an object-oriented architecture for storing objects, arranged according to "structured fields." (A0053; A0245.)

The AFP Data Stream includes as examples, scripts referred to as "Data Description Specifications" or "DDS." (A0054; A0351.) A DDS print record is defined in IBM as "the collection of fields and/or print keywords that are to be executed when the application program issues the write command. Once the record write and variable data is passed to DDS, DDS can control font, positioning, and other characteristics external to the application program." (A0054; A0352.) One such DDS keyword identified in IBM is "ZFOLD," which instructs the printer to fold the paper as it is being printed. (A0054; A0359, Fig. 62.) Notably, both IBM and the '789 Patent mention a paper folding script as an example script. (A0054; A0068 at 5:7; A0359.)

IBM describes the following as they relate to each element of the '789 patent:

- *Reading an input print data stream.* IBM describes reading "lines of output" sent from a print application and read into the AFP. (A0050; A0418.)

- *Analyzing the input data stream by means of a parser.* IBM describes that the lines of output are "optionally parsed into individual fields." (A0050; A0418.)

- *Splitting the input data into the graphically representable objects.* IBM discloses that the AFP's parser splits the lines of output into "print fields," which are graphically representable objects. An example of a print field is a bar code. (A0050, A0418.)

- *Storing the graphically representable objects in memory in an object-oriented format*. AFP is an "object-oriented" system," with the objects being stored in "printer memory." (A0050, A0231.)

- *Object-oriented format stored in the memory including at least one stored script.* AFP uses scripts referred to as "Data Description Specification" or DDS; one DDS is called ZFOLD, which is a script that instructs the printer to fold the printed paper twice, to form the shape of a "Z." (A0050, A0351, A0358-59.)

Based on these disclosures in the IBM reference as compared to the claims of the '789 patent, Samsung asserted that certain claims are either anticipated by IBM or obvious over IBM in view of various combinations of the other prior art references. (A0055.)

2. Interleaf describes the features listed in the Examiner's reasons for allowance.

Interleaf is an article that was published in June 1994, and qualifies as prior art under 35 U.S.C. § 102(b). Interleaf describes a document processing system that uses "active documents." (A0056; A0643.) Active documents are documents "built with an extensive object system." (A0056; A0643.) "Document system facilities used by this application include printing, filtering to accept CALS

compliant [a government standard] documents, and automatic hypertext linking and indexing." (A0056; A0648.)

Interleaf describes receiving a "document file stream" for printing, and parses the document to see if it has any active objects:

> If a document contains active objects within the document file stream . . . ., these become activated when the document is opened programmatically or by the user for . . . printing.

(A0056; A0652.) An example object is a "Named Graphic Object" or "NGO." (A0056; A0649.) Notably, graphic objects are example objects described both in Interleaf and in the '789 patent. (A0056; A0067 at 3:42.)

The system described in Interleaf also evaluates each object to identify if there is an associated script. "Lisp scripts can be attached to any Interleaf object, stored within or external to a document." (A0056; A0645.) An example script is an "auto-localizing" script, which references an external table to change the text to be printed according to a localized language. (A0056-57; A0646.)

Interleaf describes the following as they relate to the features of the '789 patent:

- *Reading an input print data stream*. Interleaf describes reading a "document file stream." (A0051; A0652.)
- *Analyzing the input data stream by means of a parser*. Interleaf describes an input data stream that is analyzed by means of a parser when a document contains active objects within the document file stream. (A0051; A0645; A0652-A0653; A0837-0838.)

- *Splitting the input data into the graphically representable objects*. Interleaf describes splitting out the active objects, so that each object (referred to as the "current object") can be separately evaluated. (A0051; A0653.)

- *Storing the graphically representable objects in a memory in an object-oriented format.* Interleaf states: "The system should be object-oriented to allow for easy active document building and reusability of active objects." (A0051; A0650-51.)

- *Object-oriented format stored in the memory including at least one stored script.* Interleaf describes using Lisp scripts. "Lisp scripts can be attached to any Interleaf object, stored within or external to a document." (A0051; A0645.)

In light of these disclosures in Interleaf, Samsung asserted that certain claims are either anticipated by Interleaf or obvious over Interleaf in view of various combinations of the other prior art references.

3. The Interleaf Patent describes the same system as Interleaf and includes numerous examples of scripts.

The Interleaf Patent issued on November 26, 1996, and qualifies as prior art under 35 U.S.C. § 102(b). The Interleaf Patent describes an "electronic document processing system" or EDPS that modifies documents for printing. (A0059; A0663 at 1:8-11, 2:10-11.) The Interleaf Patent describes the same system as the Interleaf article, except that it gives additional examples of scripts to trigger events during presentation or printing. A list of example events is provided in columns 6-9 of the patent. (A0059; A0665 at col. 6; A0667 at col. 9.)

In light of the Interleaf Patent, Samsung asserted that certain claims are obvious over the Interleaf Patent in view of various combinations of the other prior art references.

4. Lieberman teaches Javascript.

Lieberman is an article published in 1998, and qualifies as prior art under 35 U.S.C. § 102(b). Lieberman teaches "the use of Javascript as a scripting language," (A1009) which is recited in some of the dependent claims of the '789 Patent.

Samsung asserted that certain claims are obvious over Lieberman in view of various combinations of the other prior art references.

5. The '705 Patent teaches a print system that receives emails via a script.

The '705 Patent was filed on November 16, 1999, and qualifies as prior art under 35 U.S.C. § 102(e). The '705 Patent teaches a "printer server that receives documents as electronic mail (email) and prints a document at the appropriate printer based on the available printers, the document type, and the sender's identity." (A2043 at 2:1-4.) Specifically, the '705 Patent teaches how a script collects the email message from a folder in an archiving server and processes the email message according to the script:

> At step 121, the deposit of a message in a folder triggers the Folder: : OnMessageCreated event. As a result of the triggered event, a server-side script associated with this event is invoked and is passed the folder identification and the message identifier of the newly posted message. The script at step 122 first collects the properties of the

> message (date, sender, etc) that can be automatically derived and assigns a unique id to the document. Then, it parses the body of the message and, if the body of the message is a form, it extracts all information contained in the form.

(A2044 at 3:40-50.) (emphasis added).

Samsung asserted that certain claims are obvious over the '705 Patent in view of various combinations of the other prior art references.

C. The Examiner grants Samsung's request for *inter partes* reexamination and rejects all claims of the '789 Patent.

After considering the evidence and arguments of the parties, the Examiner issued a first office action, an Action Closing Prosecution (ACP), and a Right of Appeal Notice (RAN) interpreting key disputed terms and rejecting all claims (including all amended and new claims) of the '789 Patent. (A2165-2332.) Specifically, the Examiner rejected each of the claims for the following reasons:

1. Claim 59-64, 66, and 67 were rejected under 35 U.S.C. § 112, second paragraph, as indefinite. (A0005; A2186-2187; A2399-2400; A2438.)

2. Claims 64, 66, 67, 72, 74, and 75 were rejected under 35 U.S.C. § 314(a) as enlarging the scope of the patent being examined. (A0005: A2187-2189; A 2438.)

3. Claims 1-5, 17, 20-26, 38-41, 54, 56, 59, 61-62, 68, 70, 71, 76-78, and 80-82 were rejected under 35 U.S.C. § 102(b) as anticipated by IBM. (A0005; A2189-2195; A2439.)

4. Claims 1-7, 11-12, 17, 20-28, 32, 33, 38-44, 48, 49, 54, 56, 59, 61, 62, 68, 70, 71, and 76-82 were rejected under 35 U.S.C. § 103(a) as obvious over IBM and Interleaf. (A0005; A2196-2201; A2439.)

5. Claims 1-14, 16-18, 20-35, 37-51, 53, 54, 56, 57, 59-62, 64, 66, 68-72, 74, and 76-82 were rejected under 35 U.S.C. § 103(a) as obvious

over IBM, Interleaf, and Interleaf Patent. (A0005; A2202-2212; A2439.)

6. Claims 15, 19, 36, and 52 were rejected under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and Lieberman. (A0005-6; A2213-14; A2439.)

7. Claims 1, 2, 4-7, 11, 12, 22-23, 25-28, 32, 33, 38-39, 41-44, 48, and 49 were rejected under 35 U.S.C. § 102(b) as anticipated by Interleaf. (A0006; A2214-19; A2439.)

8. Claims 1, 2, 4-14, 16-18, 22, 23, 25-35, 37-39, 41-51, 53, 54, 57, 59, 61, 68, 70, and 76-82 were rejected under 35 U.S.C. § 103(a) as obvious over Interleaf and Interleaf Patent. (A0006; A2220-29: A2439.)

9. Claims 15, 19, 36, 52, and 58 were rejected under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and Lieberman. (A0006; A2229-2230.)

10. Claims 1-54, 56-60, 62, 64, 66, 68-71, 76, 77, and 80-82 were rejected under 35 U.S.C. § 103(a) as obvious over Interleaf Patent and IBM. (A0006; A2230-2235; A2439.)

11. Claim 55 were rejected under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and the ’705 Patent. (A0006; A2235-2238; A2439.)

12. Claims 55 and 75 were rejected under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent and ’705 Patent. (A0006; A2238-2240; A2426-2427; A2439.)

13. Claims 55, 63, and 67 were rejected under 35 U.S.C. § 103(a) as obvious over Interleaf Patent, IBM, and the ’705 Patent. (A0006; A2240-2242; A2439.)

CCP appealed the rejection to the Board. (A2445.)

D. The Board affirms the rejection of all claims of the '789 Patent, except new claims 55 and 63.

On appeal, the Board affirmed the rejection of all but two claims—new claims 55 and 63. (A0023.) Claims 55 and 63 are reproduced below.

> 55. (New) The system according to claim 54, characterized in that the data processing unit is further programmed to send and receive e-mails in the cases defined in the script.
>
> 63. (New) The printer according to claim 59, wherein said printer is further adapted to send and receive e-mails in the cases defined in the script.

(A1332 [claim 55]; A1333 [claim 63].)

The Examiner separately discussed *sending* and *receiving* emails, as recited in the claims. As for *sending* emails in cases defined in the script, the Examiner found, and the Board did not contest, that this was taught by Interleaf. (A2426; A0019.) As for *receiving* emails in cases defined in the script, the Examiner based his rejection on the '705 Patent. The Board reversed the Examiner, stating:

> The '705 Patent does not teach receiving an email via script; instead the cited portions of the '705 Patent describe a user emailing a document to a server and the server running scripts to parse the message and store the contents of the email to media. ['705 Patent] 3:17-22, 40-50. Thus the reference does not teach "receiv[ing] e-mails in the cases defined in the script" as recited by claims 55, 63, 67, and 75. Therefore, we do not sustain the Examiner's rejections of claims 55, 63, 67, and 75 under 35 U.S.C. § 103.

(A0019.) Samsung appeals the Board's failure to reject claims 55 and 63, as set forth below.

SUMMARY OF THE ARGUMENT

Claims 55 and 63 are both new dependent claims that recite "send[ing] and receiv[ing] e-mails in the cases as defined in the script." The Examiner found that Interleaf taught *sending* emails, and the '705 Patent taught *receiving* emails, as recited in these claims, and found the claims obvious over the prior art. The Board reversed the Examiner's rejection, stating that the '705 Patent does not teach *receiving* emails in the cases defined in the script. This reversal was based on a misinterpretation of the claim language. As properly construed, a script that "first collects the properties of the [email] message" from a folder in an archive server, as taught by the '705 Patent, constitutes "receiv[ing] e-mails in the cases as defined in the script."

Claim 63 should also be rejected for being indefinite for describing a printer adapted to output a print data stream (to another printer). The Examiner found claim 63's "printer adapted for the transformation of digital print data streams" to be indefinite. The '789 Patent describes two embodiments: (1) a server that outputs data to a printer, or (2) a combination printer and server (an intelligent printer). As described below, the '789 Patent never describes a *printer* outputting a digital print data stream to another printer, much less how such a system would work.

In sum, the Examiner got it right. Claims 55 and 63 are obvious under 35 U.S.C. § 103 and claim 63 is further indefinite under 35 U.S.C. § 112. The Board

erred in reversing the Examiner's rejection of these two claims. Samsung, therefore, respectfully urges the Court to reverse the Board's decision and affirm the rejection of claims 55 and 63.

ARGUMENT

I. Standards of Review.

This Court reviews the Board's fact findings for substantial evidence and reviews legal conclusions *de novo*. *In re Gartside*, 203 F.3d 1305, 1315-16 (Fed. Cir. 2000); *Tempo Lighting, Inc. v. Tivoli, LLC*, 742 F.3d 973, 976-77 (Fed. Cir. 2014). Substantial evidence is a deferential standard of review that requires the Court "to ask whether a 'reasonable mind might accept' a particular evidentiary record as 'adequate to support a conclusion.'" *Lingamfelter v. Kappos*, 513 F. App'x 934, 937 (Fed. Cir. 2012) (quoting *Dickinson v. Zurko*, 527 U.S. 150, 162 (1999)).

Obviousness is a question of law based on underlying fact findings. *Tempo Lighting*, 742 F.3d at 977 (citing *In re NTP, Inc.*, 654 F.3d 1279, 1291 (Fed. Cir. 2011)). Those fact findings include the scope and content of the prior art, the differences between the prior art and the claimed invention, the level of ordinary skill in the field of the invention, and any relevant objective considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). Thus, what a reference teaches and whether a person of ordinary skill in the art would have been motivated to combine the teachings of separate references are questions of fact that should be given great deference on appeal. *In re Gartside*, 203 F.3d at 1316; *see In re Chevalier*, 500 Fed. Appx. 932, 934 (Fed. Cir. 2013).

Claim construction is a question of law. *Tempo Lighting*, 742 F.3d at 977 (citing *In re NTP, Inc.*, 654 F.3d 1268, 1273 (Fed. Cir. 2011)). Indefiniteness is likewise a question of law reviewed *de novo*. *Teva Pharmaceuticals USA, Inc. v. Sandoz, Inc.*, 723 F.3d 1363, 1368 (Fed. Cir. 2013).

II. Claims 55 and 63 should be rejected as obvious under § 103.

A claimed invention is unpatentable if the differences between it and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art. 35 U.S.C. § 103; *C.W. Zumbiel Co. v. Kappos*, 702 F.3d 1371, 1379 (Fed. Cir. 2012). In making an obviousness determination, the Examiner must look to the prior art as a whole and determine whether, under the totality of the prior art, a person of ordinary skill in the field of invention would be motivated to combine the prior art references to make or practice the invention. *Id.*; *see KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 414 (2007).

In this case the Examiner properly applied these principles. The Board reversed the Examiner based upon the erroneous conclusion that the combination of references, in particular the '705 Patent, does not teach receiving emails in cases defined by the script, as claimed. As explained below the Board misinterpreted the "receiving" limitation in reaching its conclusion.

A. The references supporting rejections of claims 55 and 63 are properly combinable.

Claims 55 and 63 were rejected by the Examiner as being obvious over the prior art based upon multiple grounds of rejection, all of which relied upon the '705 Patent to teach their claim limitations. Specifically, as summarized in the Decision on Appeal:

> 11. Claim 55 stands rejected under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and '705 Patent. *Id.* 72-74.
>
> 12. Claims 55 and 75 stand rejected under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and '705 Patent. *Id.* 75-77; RAN 32-33.
>
> 13. Claims 55, 63, and 67 stand rejected under 35 U.S.C. § 103(a) as obvious over Interleaf Patent, IBM, and '705 Patent. *Id.* 77-79.

(A0006.)

The Board never held that the '705 Patent could not be combined with the other references identified above. Indeed, the Board confirmed the rejections of the base claims for claims 55 and 63 based on these same combinations. (A0008-0022.)

This aspect of the Board's Decision is correct. As directed to claims 55 and 63, the '705 Patent describes "a printer server that receives documents as electronic mail (email) and prints a document at the appropriate printer[.]" (A2043 at 1:66-2:4.) Accordingly, the '705 Patent, IBM, Interleaf, and the Interleaf Patent all describe well-known aspects of document printing, and are properly combinable.

"Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results" is a proper rationale for supporting a finding of obviousness under Section 103. *See KSR*, 550 U.S. at 417-20.

This appeal therefore centers on the teachings of the '705 Patent, already determined to be combinable with IBM, Interleaf, and the Interleaf Patent. As properly applied to claims 55 and 63, the '705 Patent in combination with these references supports the Examiner's obviousness rejections.

B. The '705 Patent teaches "receiv[ing] emails in the cases defined in the script."

Claims 55 and 63 are reproduced below.

> 55. (New) The system according to claim 54, characterized in that the data processing unit is further programmed to send and receive e-mails in the cases defined in the script.
>
> 63. (New) The printer according to claim 59, wherein said printer is further adapted to send and receive e-mails in the cases defined in the script.

(A1332 [claim 55]; A1333 [claim 63].)

At the Patent Office, the Examiner separately discussed *sending* and *receiving* emails, as recited in the claims. As for *sending* emails in cases defined in the script, the Examiner found, and the Board did not contest, that this was taught by the Interleaf Patent. (A02426; A0019.) And, indeed, the Interleaf Patent expressly teaches, for instance, a "WYSIWYG document that can send itself over an electronic mail system." (A0668 at 7:50-51 and at 8:19-22.)

As for *receiving* emails in cases defined in the script, the Examiner based his rejection on the '705 Patent. The Examiner explained:

"The '705 patent shows that it was known in the art to receive an email message via a script:

> [T]he user sends the document to be archived as an attachment to an email addressed to the public folder that should contain the document, e.g., abc@saveme.com, where abc is the name of a public folder on the archiving server saveme.com 120 …. The deposit of a message in a folder triggers the Folder: :OnMessageCreated event. As a result of the triggered event, a server-side script associated with this event is invoked and is passed the folder identification and the message identifier of the newly posted message. The script at step 122 first collects the properties of the message (date, sender, etc.) that can be automatically derived and assigns a unique id to the document. Then, it parses the body of the message and, if the body of the message is a form, it extracts all information contained in the form. '705 patent at 3:17-22, 40-50.

The PO next argues that 'in the '705 Patent, a script is triggered by an email. In contrast, claim 55 of the '789 Patent recites the causal opposite—an email is triggered by a script.' (PO Remarks at 20, emphasis in original). Again, the '705 patent is being used to show that it was known in the art to receive an email message via a script."

(A2425.)

The Board reversed the Examiner, stating:

> The '705 Patent does not teach receiving an email via script; instead the cited portions of the '705 Patent describe a user emailing a document to a server and the server running scripts to parse the

> message and store the contents of the email to media. ['705 Patent] 3:17-22, 40-50. Thus the reference does not teach "receiv[ing] e-mails in the cases defined in the script" as recited by claims 55, 63, 67, and 75. Therefore, we do not sustain the Examiner's rejections of claims 55, 63, 67, and 75 under 35 U.S.C. § 103.

(A0019.)

Implicit in the Board's finding is a presumption that in order to receive an email in the cases defined in the script as that term is used in claims 55 and 63, the system (claim 55) or printer (claim 63) must actually initiate the email.

The Board erred in this finding. To the extent the Board is limiting the claims to require that the system (claim 55) or printer (claim 63) *initiate* the email, such is not required by the "receiving" language of the claims. Instead, as properly interpreted, receiving an email in cases defined in the script includes collecting an email from an archive server for processing in cases defined in the script, as taught in the prior art '705 Patent.

During an *inter partes* reexamination proceeding, "claims ... are to be given their broadest reasonable interpretation consistent with the specification, and ... claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech. Ctr.,* 367 F.3d 1359, 1364 (Fed.Cir.2004) (internal quotation marks and citation omitted). Nothing in the plain language of claims 55 or 63 imposes a limitation requiring that the printer or system initiate the received email. On the contrary,

these claims provide only that the system or printer is programmed/adapted to "send and receive e-mails in the cases defined in the script." *See supra*. There is no support in the specification that defines or otherwise restricts *how* an email would be received in the cases defined in the script. The bare language of the '789 patent merely states: "Systems that operate on the method according to the invention, such as printing systems, are capable of sending and receiving emails[.]" (A2045 at 5:38-40.) In this context, receiving emails in cases defined in the script can only mean periodic receipt of emails via a script from some location—the exact way the '705 Patent does it. Nothing about the claim language requires a script be used to initiate the email. Imposing this requirement would be tantamount to re-writing claims 55 and 63 and impermissibly limiting their scope. *Cf. Tempo Lighting, Inc. v. Tivoli, LLC*, 742 F.3d 973, 977 (Fed. Cir. 2014) (affirming broadest reasonable claim construction).

As pointed out by the Examiner, the '705 Patent teaches a multi-step process, and the second step reads on the claimed "receiving" language exactly. The '705 Patent teaches that a user sends an email, and that the system stores the email in an archiving server. (A2044 at 3:17-22.) The next step is important, because it reads exactly on the claims. After the email message has been stored in the archive server, the '705 Patent system collects (receives) the emails from the archive server in cases defined in a script. The '705 Patent teaches: "The script at

step 122 first collects the properties of the message (date, sender, etc) …. Then, it parses the body of the message[.]" (A2044 at 3:45-48.) (emphasis added). The script's "collect[ion]" of the message cited above corresponds to the claimed "receiving e-mails in cases defined in the script" as recited by claims 55 and 63. That is, after the email has been stored, it is the script that causes the stored email to be collected/received in cases defined in the script, and further acted upon by the script. Accordingly, a script that "first collects the properties of the [email] message" from a folder in an archive server, as taught by the '705 Patent, constitutes "receiv[ing] e-mails in the cases as defined in the script."

Considering the prior art references as a whole, and under a proper claim interpretation, it was error for the Board to reverse the Examiner's rejection of claims 55 and 63. These two claims are obvious in light of the prior art and should be rejected.

III. Claim 63 should be rejected under §112.

As determined by the Examiner, new claim 63 should be rejected on the additional ground that it is indefinite under 35 U.S.C. § 112, ¶2 because it fails to teach ". . . wherein said printer is adapted . . . to output print data stream." (A2399-2400; A0008; A2187.)

The Patent Act requires that a patent specification "conclude with one or more claims particularly pointing out and distinctly claiming the subject matter

which the applicant regards as [the] invention." 35 U.S.C. § 112, ¶ 2 (2006 ed.). The Federal Circuit has traditionally held that a patent claim passes the § 112, ¶ 2 threshold so long as the claim is "amenable to construction," and the claim, as construed, is not "insolubly ambiguous." *Biosig Instruments, Inc. v. Nautilus, Inc.*, 715 F.3d 891, 898–899 (Fed. Cir. 2013), *rev'd*, 134 S.Ct. 2120 (2014). But on June 2, 2014, the Supreme Court rejected the "insolulably ambiguous" standard, which it criticized as "tolerat[ing] some ambiguous claims but not others," and held that "a patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S.Ct. 2120, 2124 (2014). The linchpin of the standard articulated in *Nautilus* is certainty.

Although *Nautilus* had not yet issued at the time the Examiner issued its Action Closing Prosecution, the Examiner's decision to reject claims 59 and 64, and their dependent claims, including claim 63, as indefinite reflects this desire for certainty. (The Board confirmed the rejection of claims 59 and 64 for different reasons.) Specifically, the Examiner found the claims indefinite because the '789 Patent fails to teach ". . . wherein said printer is adapted . . . to output print data stream." (A2399-2400; A0008; A2187.) When read in the context of the patent specification, the scope of claim 59 and dependent claim 63 is not clear. One of

ordinary skill in the art would not understand what is meant by "a printer that outputs a printer data stream," and, the specification does not contain any disclosure relating to this limitation.

The specification refers to a "printer," but only discusses "output[ting] said combined output print data stream" *to* a printer (from a print server). More specifically, the specification of the '789 Patent describes two different embodiments: (1) a print server connected to a printer and (2) an intelligent printer. (A0069 at 8:24-25 – A0070 at 9:11-12.) These two embodiments are mutually exclusive so that the device 3 (A0028 at FIG. 1) can **either** be "a computer such as a PC [printer server] **or else** an intelligent output device such as an intelligent printer." (A0069 at 8:59-61.) (emphasis added). Nothing in the '789 Patent teaches that the device 3 can be *both* a printer server *and* an intelligent printer. Nor does it explain how or even that a printer can output a data stream.

In reversing the Examiner's rejection of claim 63 as indefinite, the Board found that the specification "describes device 3 as either a computer or an intelligent printer in the same passage where it describes device 3 as outputting a print data stream to a device that is 'preferably a printer 9.'" (A0008 (citing '789 Patent at 8:59-9:34).) However, the fact that these descriptions are in the same passage does not negate the "or else" language discussed and cited above.

Therefore, new claim 63 is indefinite, and must be rejected under section 112. *See ePlus, Inc. v. Lawson Software, Inc.*, 700 F.3d 509, 519 (Fed. Cir. 2012).

CONCLUSION AND PRAYER

For the foregoing reasons, Samsung urges the Court to reverse the Board's decision reversing the Examiner's rejection of claims 55 and 63 and order claims 55 and 63 rejected.

Respectfully Submitted,

/s/ David L. McCombs

David L. McCombs
Debra J. McComas
HAYNES AND BOONE, LLP
2323 Victory Avenue, Suite 700
Dallas, Texas 75219
Telephone: 214.651.5375
Facsimile: 214.200.0525
David.McCombs@haynesboone.com
Debbie.McComas@haynesboone.com

Attorneys For Appellant,
Samsung Electronics Corporation, Ltd.

ECF CERTIFICATION

I hereby certify that (i) the required privacy redactions have been made pursuant to Federal Rule of Civil Procedure 5.2; (ii) the electronic submission is an exact copy of the paper document; (iii) the document has been scanned for viruses with the most recent version of a commercial virus scanning program and is free of viruses; and (iv) the paper document will be maintained for three years after the mandate or order closing the case issues.

/s/ Debra J. McComas
Debra J. McComas

CERTIFICATE OF COMPLIANCE

1. This brief complies with the type-volume limitation of FED. R. APP. P. 32(a)(7)(B) because:

■ this brief contains 6,110 words, excluding the parts of the brief exempted by FED. R. APP. P. 32(a)(7)(B)(iii).

2. This brief complies with the typeface requirements of FED. R. APP. P. 32(a)(5) and the type style requirements of FED. R. APP. P. 32(a)(6) because:

■ this brief has been prepared in a proportionally spaced typeface using Microsoft Word 2003 in 14 point Times New Roman font.

/s/ Debra J. McComas
Debra J. McComas

2274819_8

FORM 30. Certificate of Service

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

CERTIFICATE OF SERVICE

I certify that I served a copy on counsel of record on Jun 20, 2014 by:

- ☐ US mail
- ☐ Fax
- ☐ Hand
- ☒ Electronic Means
  (by email or CM/ECF)

David L. McCombs
Name of Counsel

/s/ David L. McCombs
Signature of Counsel

Law Firm Haynes and Boone, LLP
Address 2323 Victory Avenue, Suite 700
City, State, ZIP Dallas, Texas
Telephone Number 214.651.5375
FAX Number 214.200.0525
E-mail Address David.McCombs@haynesboone.com

NOTE: For attorneys filing documents electronically, the name of the filer under whose log-in and password a document is submitted must be preceded by an "/s/" and typed in the space where the signature would otherwise appear. Graphic and other electronic signatures are discouraged.

ADDENDUM

A. Decision on Appeal (R0002-0023)

B. United States Patent No. 6,684,789 (R0064-0072)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CORP. LTD.
Requester and Respondent

v.

CCP SYSTEMS AG
Patent Owner and Appellant

Appeal 2013-009045
Reexamination Control 95/001,398
Patent 6,684,789 B2
Technology Center 3900

Before HOWARD B. BLANKENSHIP, STANLEY M. WEINBERG, and STACEY G. WHITE, *Administrative Patent Judges*.

WHITE, *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF THE CASE

A. Introduction

This is an appeal from the Examiner's decision in an *inter partes* reexamination of U.S. Patent 6,684,789 B2 ("the '789 Patent"), owned by CCP Systems AG ("Appellant"). This reexamination proceeding arose from a third party request filed by Samsung Electronics Corp. ("Requester"). After an Action Closing Prosecution ("ACP"), the Examiner issued a Right of Appeal Notice ("RAN"). The RAN lists claims 1-64, 66-72, and 74-82 as finally rejected. Claims 1-53 are original patent claims. Claims 54-64, 66-72, and 74-82 are proposed new claims. Proposed claims 65 and 73 have been canceled. RAN 2. Patent Owner invokes our review of rejected claims under 35 U.S.C. § 134(b) (2002). An oral hearing was held October 9, 2013. We have jurisdiction under 35 U.S.C. §§ 134(b) and 315 (2002).

We AFFIRM-IN-PART.

B. Related Proceedings

Appellant has informed us that there are no related appeals or interferences before the Board. App. Br. 2. Appellant has also informed us that *CCP Systems AG* v. *Samsung Electronics Corp., Ltd. Samsung Electronics America, Inc., Samsung Networks, Inc. and IBM Corp.,* 2:09-cv-04254 (D. N.J.) is related to this reexamination proceeding. *Id.* 65. The patent portion of that litigation has been stayed. *Id.*

C. The Invention

The '789 Patent is directed to apparatuses and methods to transform a digital print data stream. '789 Patent 1:6-7, Abstract. In the prior art print

systems, an application controls a print driver that converts print data into a format (page description language or "PDL") suitable for the printer. *Id.* 1:9-16, 2:52-61. The invention of the '789 Patent sets out to handle more complex page description language at a higher level of abstraction by parsing an input print data stream for graphically representable objects, storing and transforming these objects, and outputting an output print data stream. *Id.* 2:62-3:20.

Claim 1 is illustrative of the appealed subject matter and is reproduced below:

> 1. A method for the transformation of digital print data streams, in which
>
> (i) an input print data stream (2) is read in,
>
> (ii) this is analyzed by means of a parser (4) for graphically representable objects (5) and is split up into these graphically representable objects (5), and
>
> (iii) the graphically representable objects (5) are stored in a memory (6) in an object-oriented format, and
>
> (iv) the graphically representable objects (5) stored in the memory (6) in an object-oriented format are transformed into a format for the control of an output device (9), preferably a printer, and
>
> (v) the objects thus transformed are combined into an output print data stream (10) and are output,
>
> characterized in that graphically representable objects (5, 5a) are stored in the memory (6) in an object-oriented format, to which at least one stored script is assigned, which is executed in the cases defined in the script.

D. The Prior Art

Pelletier ("Interleaf Patent")	U.S. 5,579,519	Nov. 26, 1996
Berchtold ("'705 Patent")	U.S. 6,678,705 B1	Jan. 13, 2004

AS/400 Guide to Advanced Function Presentation and Print Services Facility, Fourth Edition, IBM Printing Systems, January 1999 ("IBM").

English, Paul M. et al, Interleaf Active Documents, 7(2) Electronic Publishing, 75-87, June 1994 ("Interleaf").

Lieberman, Henry, "Integrating user interface agents with conventional applications," Knowledge-Based Systems 11 (1998), pp 15-23 ("Lieberman").

E. The Rejections

1. Claims 59-64, 66, and 67 stand rejected under 35 U.S.C. § 112, second paragraph, as indefinite. ACP 23-24; RAN 2-6, 44.

2. Claims 64, 66, 67, 72, 74, and 75 stand rejected under 35 U.S.C. § 314(a) as enlarging the scope of the patent being reexamined. ACP 24-26; RAN 6, 44.

3. Claims 1-5, 17, 20-26, 38-41, 54, 56, 59, 61-62, 68, 70, 71, 76-78, and 80-82 stand rejected under 35 U.S.C. § 102(b) as anticipated by IBM. ACP 26-32.

4. Claims 1-7, 11-12, 17, 20-28, 32, 33, 38-44, 48, 49, 54, 56, 59, 61, 62, 68, 70, 71, and 76-82 stand rejected under 35 U.S.C. § 103(a) as obvious over IBM and Interleaf. *Id.* 33-38.

5. Claims 1-14, 16-18, 20-35, 37-51, 53, 54, 56, 57, 59-62, 64, 66, 68-72, 74, and 76-82 stand rejected under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, and Interleaf Patent. *Id.* 39-49.

6. Claims 15, 19, 36, and 52 stand rejected under 35 U.S.C.

§ 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and Lieberman.[1] *Id.* 50-51.

7. Claims 1, 2, 4-7, 11, 12, 22-23, 25-28, 32, 33, 38, 39, 41-44, 48, and 49 stand rejected under 35 U.S.C. § 102(b) as anticipated by Interleaf. *Id.* 51-56.

8. Claims 1, 2, 4-14, 16-18, 22, 23, 25-35, 37-39, 41-51, 53, 54, 57, 59, 61, 68, 70, and 76-82 stand rejected under 35 U.S.C. § 103(a) as obvious over Interleaf and Interleaf Patent. *Id.* 57-66.

9. Claims 15, 19, 36, 52, and 58 stand rejected under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and Lieberman. *Id.* 66-67.

10. Claims 1-54, 56-60, 62, 64, 66, 68-71, 76, 77, and 80-82 stand rejected under 35 U.S.C. § 103(a) as obvious over Interleaf Patent and IBM. *Id.* 67-72.

11. Claim 55 stands rejected under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and '705 Patent. *Id.* 72-74.

12. Claims 55 and 75 stand rejected under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and '705 Patent. *Id.* 75-77; RAN 32-33.

13. Claims 55, 63, and 67 stand rejected under 35 U.S.C. § 103(a) as obvious over Interleaf Patent, IBM, and '705 Patent. *Id.* 77-79

[1] Appellant does not separately challenge the rejections directed to dependent claims 15, 19, 36, 52, and 58. App. Br. 5 n 5. Thus, we will not specifically address these claims outside of our conclusion.

F. Issues Presented

The briefing in response to the Examiner's RAN presents us with the following issues:

1. Did the Examiner err in finding indefinite a printer "adapted to transform the graphically representable objects into a format for the control of an output device, to combine the objects into an output print data stream, and to output said combined output print data stream" as recited in claims 59 and 64?

2. Did the Examiner err in concluding that claims 64 and 72 enlarge the scope of the claims of the '789 Patent?

3. Did the Examiner err in finding IBM to disclose (a) analyzing an input print data stream by a parser and (b) assigning a script to an object?

4. Did the Examiner err in finding Interleaf to disclose transforming, reading, and parsing a print data stream?

5. Did the Examiner err in finding the cited references to teach a data stream "formatted in page description language" as recited by claim 76?

6. Did the Examiner err in finding the cited references to teach a "parser [that] is a syntax analyzer" as recited by claim 77?

7. Did the Examiner err in finding the cited references to teach a printer?

8. Did the Examiner err in combining IBM, Interleaf, and Interleaf Patent?

9. Did the Examiner err in combining IBM, Interleaf, Interleaf Patent, and '705 Patent?

10. Did the Examiner err in finding no nexus between the '789 Patent and Appellant's evidence of commercial success?

II. ANALYSIS

A. Are Claims 59 and 64 Indefinite?

The Examiner rejected claims 59 and 64 as indefinite because the Specification did not teach a printer adapted to output a print data stream. ACP 24; RAN 5-6. Appellant argues that support for the claimed printer is found in Figure 1 of the '789 Patent that shows device 3 outputting a print data stream to printer 9. '789 Patent 8:57-9:12, Fig. 1. Device 3 is "a computer such as a PC or else an intelligent output device such as an intelligent printer – which operates in accordance with the method according to the present invention." *Id.* 8:59-61. Requester avers the Specification does not describe an embodiment where a printer outputs a print data stream to printer 9. Resp. Br. 3. We disagree. The Specification describes device 3 as either a computer or an intelligent printer in the same passage where it describes device 3 as outputting a print data stream to a device that is "preferably a printer 9." '789 Patent 8:59-9:34. Thus, we find that Figure 1 and its supporting text describe an inventive printer that outputs a print data stream. Therefore, we do not sustain the Examiner's rejection under 35 U.S.C. § 112, second paragraph of claims 59 and 64 and their dependent claims 60-63, and 66-67.

B. Do Claims 64 and 72 Enlarge the Scope of the '789 Patent Claims?

The Examiner rejected claims 64 and 72 under 35 U.S.C. § 314(a), which provides that "the patent owner shall be permitted to propose … a new claim or claims, except that no proposed … new claim enlarging the

scope of the claims of the patent shall be permitted." 35 U.S.C. § 314(a) (2002). Claims 64 and 72 are directed to "a printing system" that includes a printer or print server. The Examiner found the new claims improper because there are no original claims directed to "a printing system." RAN 6. Appellant argues that original claim 1 includes every limitation found in claims 64 and 72. App. Br. 12. In addition, original claim 17 recites "a system" with a data processing unit programmed to perform the method of claim 1 and original claims 20 and 21 further recite a printer or print server that has a system "as claimed in claim 17." *Id.*; '789 Patent 10:57-62, 11:13-18.

In a reexamination, "[a] new claim enlarges if it includes within its scope any subject matter that would not have infringed the original patent." *Thermalloy, Inc. v. Aavid Eng'g, Inc.,* 121 F.3d 691, 692 (Fed. Cir. 1997) (citing *Quantum Corp. v. Rodime, PLC*, 65 F.3d 1577, 1580 (Fed.Cir.1995)). Requester argues that an accused system performing any of claim 1's steps using logic hardware as opposed to a preprogrammed data processing unit would not fall within the purview of the original claims, but potentially would be captured by new claims 64 and 72. Resp. Br. 4. Appellant avers that the new claims recite a data processing unit and thus, the data processing unit is required to perform all data processing steps. Rebuttal Br. 6.

We agree with Requester. The newly added claims are broader in that a device "adapted to" perform certain steps encompasses more than the originally claimed "data processing unit [that] is programmed" to perform certain steps. The original claims require that the data processing unit

specifically be programmed to carry out the claimed method. Claims 64 and 72, however, have no such requirement. Under the new claims, an accused device with a data processing unit that is not programmed to perform all of the claimed steps could still infringe so long as something else in the device is "adapted to" perform the missing step. Thus, we sustain the Examiner's rejection under 35 U.S.C. § 314(a) of claims 64 and 72 and their dependent claims 66, 67, 74, and 75.

C. Does IBM Disclose Analyzing an Input Print Data Stream by a Parser and Assigning a Script to an Object?

Appellant argues the Examiner erred in determining that the IBM reference anticipates the claims of the '789 Patent. App. Br. 12-13. Appellant asserts that IBM does not disclose an embodiment that both parses an input stream and assigns a script to objects parsed from that input stream as is required by independent claims 1, 22, 38, 54, and 68. *Id.* 13-14.

According to Appellant, IBM discloses two embodiments. *Id.* 13. In the first embodiment, an application program describes the formatting of the document to be printed ("program-described"). *Id.* (citing IBM 25). In the second embodiment, an application outputs raw data with the formatting to be "externally-described" by a data description specification ("DDS"). *Id.* Appellant argues that the Examiner found DDS to disclose the claimed script, but that DDS is never applied to "program-described" output. Thus, the Examiner improperly relied on parsing found in the "program-described" embodiment and a script found in the "externally-described" embodiment. Requester asserts that Appellant is taking a narrow view of IBM's

disclosures and that the "program-described" and "externally-described" functionalities are not mutually exclusive. Resp. Br. 5.

IBM describes "DDS as an enabler for document and reports … [that] provides the application programmer with the capability to produce very customized output," but notes that "there are environments where this tight integration is less desirable … [because] it can make the task of coding application logic and output logic more complex because the logic is intertwined." IBM 193. IBM then describes "[n]ew output formatting objects …, page definitions and form definitions" that "provide a means to separate page formatting from the application program." *Id.*

The Examiner points to IBM's discussion of splitting output into "print fields" as disclosing the claimed parser. ACP 27 (citing IBM 194). That portion of IBM describes using a page definition so that "[i]nput print lines are read in, optionally parsed into individual fields, and place[d] on the page." IBM 194. However, "page definition is an alternative to DDS for formatting data on a page independent of the application program." IBM 37. Thus, Appellant argues that page definitions and DDS are distinct alternative embodiments. Requester asserts that DDS scripts may be applied to the parsed data through the application of an overlay as shown in Figure 116 of IBM. Resp. Br. 5 (annotated Fig. 116). This, however, is not supported by the reference. IBM describes using form definitions in conjunction with page definitions to supply and place an overlay on the page. IBM 195, 214. We have not been directed to persuasive evidence that DDS is ever applied to data that has been parsed with a page definition. Thus, we do not sustain

the Examiner's rejection under 35 U.S.C. § 102 of claims 1-5, 17, 20-26, 38-41, 54, 56, 59, 61, 62, 68, 70, 71, 76-78, and 80-82 as anticipated by IBM.

D. Does Interleaf Disclose Transforming, Reading, and Parsing a Print Data Stream?

Appellant asserts that the Examiner erred in concluding that Interleaf anticipates the claims of the '789 Patent. Specifically, Appellant argues that Interleaf does not disclose transforming, reading, or parsing a print data stream as required by independent claims 1, 22, 38, 54, and 68. App. Br. 24.

Interleaf is "[a] commercial structured document processing system" for "the design, implementation, and delivery of active documents." Interleaf 75. Interleaf's documents include component objects that contain content and document formatting. *Id.* 76. One of Interleaf's purported advantages is its ability to use external publishing functionality including document printing. *Id.* 82. In Interleaf, "active objects within the document file stream become activated … when the document is opened programmatically or by the user for viewing, editing, or printing." *Id.* 84.

Appellant argues that Interleaf does not disclose a print data stream. App. Br. 24. The Examiner construed "digital print data stream" to include on "a 'digital' 'data stream' where the stream may be sourced from a 'document' and may or may not be output to a 'printer.'" ACP 148. The Specification does not define this term and we see no error in the Examiner's reasonable construction. Pursuant to this construction, the Examiner found that Interleaf's "document file stream" discloses the claimed "digital print data stream." *Id.* The Examiner found that Interleaf's document file stream, which is sourced from an Interleaf document, may be printed and that it is

parsed for active objects. *Id.* 51. Dr. David Birnbaum testified on behalf of Appellant that Interleaf's file stream is merely what is read in when an Interleaf document is opened and that while Interleaf conceivably may create a print data stream no such print data stream is discussed in the reference. Birnbaum Decl. ¶¶ 100-03. According to Dr. Birnbarum, a print data stream would not exist until after a user clicks a print icon. *Id.* ¶ 102. We are not persuaded by Dr. Birnbaum's testimony because it fails to provide convincing evidence that an Interleaf document file stream becomes a distinct print data stream once a user presses a print button. We agree with the Examiner's reasonable findings and we concur that Interleaf's "document file stream" falls within the broadest reasonable interpretation of the claimed "print data stream."

The remaining question is whether Interleaf discloses reading, parsing, and transforming a print data stream. As to the transforming step, the Examiner points to Interleaf's discussion of changing a document's style in the middle of a text stream, changing objects in the stream, and programmatically applying different graphics properties such as changes to color, size, or position of an object. ACP 52 (citing Interleaf 76, 81-84). Appellant argues that these disclosures refer to editing or creating a document and not printing the document. App. Br. 25. We disagree with Appellant. As noted above, the print data stream as broadly construed includes Interleaf's document file stream. Thus, we agree with the Examiner's finding that Interleaf discloses transforming a print data stream.

As to the "reading in" step, the Examiner found this to be disclosed by Interleaf's discussion of opening a document for viewing, editing, or

printing. ACP 51-52. Appellant argues that this is akin to opening a document in Word and is completely unrelated to the claimed reading in of a print data stream. App. Br. 26. Here again, Appellant's view of a print data stream is too narrow. We agree with the Examiner's finding that Interleaf's discussion of opening a document discloses the claimed "reading in" of a print data stream.

Finally, as to the parsing step the Examiner looks to Interleaf's discussion of a Lisp interpreter that analyzes the data stream by means of a parser. ACP 52. Appellant argues that the Lisp interpreter only interprets Lisp scripts and "it is incapable of interpreting print data streams." App. Br. 27; Birnbaum Decl. ¶ 105. The Examiner, however, points out that Interleaf discloses binding the Lisp code and the underlying document data into a single document file stream. ACP 52; *see also* Interleaf 77 ("Lisp scripts can be attached to any Interleaf object, stored within or external to a document."). As such, the Lisp interpreter operates on the Interleaf document file stream. Thus, we are persuaded that Interleaf discloses parsing of a print data stream. Therefore, we a sustain the Examiner's rejection under 35 U.S.C. § 102 of claims 1, 2, 4-7, 11, 12, 22, 23, 25-28, 32-33, 38-39, 41-44, 48, and 49 as anticipated by Interleaf.

E. Do the Cited References Teach a Data Stream "Formatted in Page Description Language"?

Claim 76 depends from claim 1 and further limits the input data stream by requiring it to be "formatted in a page description language." The Examiner found this limitation to be taught by IBM's disclosure of Postscript files and also by the disclosure of Interleaf frames found in both

Interleaf and Interleaf Patent. ACP 31, 64. Appellant disagrees with both findings. App. Br. 20, 27.

As to the IBM reference, Appellant argues that the Examiner did not examine this limitation in the broader context of the claim. App. Br. 20. In particular, claim 1 requires that an input data stream be assigned a script. The Examiner's rejection would apply DDS to the Postscript file, however, "a Postscript file would *already* be formatted and not require or allow a DDS file to format it." *Id.* (citing IBM 25). IBM states that "document data streams, such as Postscript … can be converted to AFP and then stored or printed from the AS/400." IBM 8. In addition, IBM teaches that an AFP file can contain DDS keywords and that AFP functions include printing graphical information from DDS. Resp. Br. 9; IBM 26. Thus, Requester argues that IBM teaches applying DDS to a Postscript file. Resp. Br. 9. We agree with Requester and find that the Examiner did not err in finding that one of ordinary skill in the art would read the IBM reference to teach the limitations of claim 76.

As to Interleaf and Interleaf Patent, Appellant argues that the Examiner erred because "Interleaf frames are merely the manner in which objects are depicted on a screen; they have nothing to do with printing, and certainly do not constitute an input print data stream." App. Br. 27. As previously discussed, print data streams encompass the Interleaf's document file stream. *Supra* pp. 11-12. Frames are a component of that document file stream. Interleaf 76. Thus, the Examiner did not err in finding the Interleaf references to teach the limitations of claim 76.

F. Do the Cited References Teach a "Parser [That] Is a Syntax Analyzer"?

Claim 77 depends from claim 76 and further requires the parser to be a syntax analyzer. The Examiner found this limitation to be taught by IBM's disclosure of assigning a DDS script to an object and by the Lisp interpreter disclosed in the Interleaf references. ACP 31, 64. As to the IBM reference, Appellant argues that DDS is not applied to a Postscript file. App. Br. 20. We disagree for reasons discussed above. *Supra* p. 12. As to the Interleaf references, the Appellant argues that the Lisp interpreter analyzes the Lisp scripts and not the input data stream. We disagree for reasons previously discussed. *Supra* p. 13. Thus, the Examiner did not err in finding that IBM and the Interleaf references teach the limitations of claim 77.

G. Do the Cited References Teach a Printer?

Claims 20, 59, and 64 each are directed to a printer. These claims stand rejected over IBM's teachings as combined with the teachings of Interleaf and/or Interleaf Patent. Appellant argues that neither IBM nor the Interleaf references teach a printer. App. Br. 21-22, 28. Appellant contends that the AS/400 described in the IBM reference is not a printer; instead it merely communicates with printers. App. Br. 21. Further, Appellant maintains that being attached to a printer via a cable or network does not make as the AS/400 a printer nor does it include a printer. *Id.* 22. Requester argues that the claims do not require a single chassis and thus, IBM's disclosure of an AS/400 connected to a printer teaches the recited printers. Resp. Br. 9 (citing RAN 11). Appellant responds that even under the

broadest reasonable interpretation a printer must be a single device and not a distributed system. Rebuttal Br. 17; *see also* App. Br. 22 n. 14. We disagree. Appellant has not directed us to persuasive evidence in the Specification that would limit the recited printer to a single chassis. Thus, we see no error in the Examiner's finding that IBM teaches a printer.

Claim 59, which is directed to a printer, stands rejected as obvious over the Interleaf references. The Examiner found that "the Interleaf patent is directed towards 'a Document Processing System for creating, editing, printing and presenting active electronic documents.' Thus, the 'system' of the Interleaf patent is for 'printing,' and can therefore be considered a 'printer.'" RAN 10. Appellant argues that this is insufficient to render obvious the claimed printer. App. Br. 28. We agree. The Examiner has not provided sufficient evidence that one of ordinary skill in the art would learn the claimed printer from the Interleaf references. Those references merely state that Interleaf is "for printing" without any disclosure as to a printer. Thus, the Examiner erred in rejecting claim 59 under 35 U.S.C. § 103 over Interleaf and Interleaf Patent.

H. Did The Examiner Err in Combining IBM, Interleaf, and Interleaf Patent?

Appellant argues that the Examiner erred in combining the teachings of IBM, Interleaf, and Interleaf Patent. App. Br. 29-34. Appellant asserts that "the Examiner defined the field of the invention broadly, inconsistently, and artificially, and then tried to shoehorn the prior art into it." *Id.* 31. According to Appellant the Examiner found that the '789 Patent's field of art to be "object-oriented document publishing systems with scripting

functionality." *Id.* (citing ACP 33). This, however, misstates the Examiner's finding. The Examiner was discussing the analogous nature of IBM and the Interleaf references. ACP 33, 69. According to the Examiner, it would have been obvious to modify IBM with the teaching of Interleaf as related to assigning a script to be executed. *Id.* 33, 69-70. The Examiner further found that "Interleaf suggests the combination because it is explicitly designed to leverage outside printing and publishing functionality." *Id.* 34.

Appellant also contends that the Examiner was incorrect in finding that "documents produced by IBM are the same type of documents that the Interleaf Patent's EDPS processes and prints." App. Br. 32 (citing ACP 70). Appellant states, without citation to the reference, that Interleaf's active documents must be converted to inactive documents before they may be printed. *Id.* 32. Requester argues that this is contrary to the record because in Interleaf, "active objects within the document file stream become activated … when the document is opened programmatically or by the user for viewing, editing, or printing." Resp. Br. 12 (quoting Interleaf 84).

The Examiner found that the combination of IBM's printing process and Interleaf Patent's print-scripting was a combination of well-known techniques to yield predictable results. ACP 69; *see also* ACP 34-35 (making same argument in regards to the combination of Interleaf and IBM). Our reviewing court has reaffirmed that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Leapfrog Enter., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1161 (Fed. Cir. 2007) (quoting *KSR Int'l v. Teleflex Inc.,* 550 U.S. 398, 416 (2007)). We find the Examiner's reasoning

to be rational and well supported by the factual record. Accordingly, we find Appellant's arguments unpersuasive that the cited references have been improperly combined by the Examiner.

I. Does the '705 Patent Teach Receiving Email Via a Script ?

Claims 55, 63, 67, and 75 each recite a device adapted or programmed "to send and receive e-mails in the cases defined in the script." The Examiner cites the '705 Patent as teaching receiving email messages via a script. RAN 32. Appellant argues that this limitation is not taught by the '705 reference. App. Br. 40-41. Requester asserts that the '705 Patent shows that the disputed limitation was known in the art. Resp. Br. 14. We agree with Appellant. The '705 Patent does not teach receiving an email via script; instead the cited portions of the '705 Patent describe a user emailing a document to a server and the server running scripts to parse the message and store the contents of the email to media. *Id.* 3:17-22, 40-50. Thus, this reference does not teach "receiv[ing] e-mails in the cases defined in the script" as recited by claims 55, 63, 67, and 75. Therefore, we do not sustain the Examiner's rejections of claims 55, 63, 67, and 75 under 35 U.S.C. § 103.

J. Is There a Nexus Between the '789 Patent and Appellant's Evidence of Commercial Success?

Appellant argues that its evidence of commercial success outweighs any prima facie case of obviousness. App. Br. 35-36. The Examiner disagreed and found there to be no nexus between Appellant's evidence and the '789 Patent. RAN 20.

Appellant provides declarations from its CEO, Roland Widuch, and Director of Business Development, Christoph Picht. Widuch testifies that IBM took an exclusive license to Appellant's JScribe® software. Widuch Decl. ¶ 7. The term "JScribe" is listed on the face of the '789 Patent. '789 Patent 5:56-59 ("systems operating by the method according to the invention are also designated JScribe (registered trademark) systems and, accordingly, the method according to the invention is also designated JScribe."); *see also* 6:38-41, 6:66-67, 7:57-59, 8:26-34. Picht testifies that JScribe® embodies at least claims 1-15 of the '789 Patent. Picht Decl. ¶¶ 11, 15. Widuch further testifies that IBM provided a sublicense to Konica-Minolta Business Solutions, Inc. Widuch Decl. ¶ 8. In addition, Widuch states that Requester took a sublicense and agreed to pay IBM a lump sum and a running royalty for each device that includes JScribe® technology. *Id.* ¶ 10. IBM and Requester issued a joint press release announcing the launch of printers and multifunction products using JScribe®. *Id.* ¶ 11. Requester issued a press release and a white paper discussing its use of JScribe®. *Id.*¶¶ 15-16.

Requester argues that Appellant's evidence lacks the required nexus. Resp. Br. 13. Requester asserts that the licenses in question do not specifically mention the '789 Patent. *Id.* In addition, the licenses were for the source code of JScribe® and source code provides a value and a commercial benefit in itself. *Id.* Further, Requester avers that the Picht declaration is conclusory. *Id.* We agree with Requester. Picht's declaration is conclusory in that it merely repeats the limitations of the claim without providing further analysis as to JScribe's® functionality. In addition, none of the licenses relied upon by Appellant are of record and the testimony

provided regarding these licenses fails to state that the '789 Patent is specifically mentioned in any of the agreements. Thus, we find that the Examiner did not err in finding there to be a lack of nexus between Appellant's evidence of commercial success and the claims of the '789 Patent.

III. CONCLUSION

To summarize, our decision is as follows.

- The Examiner's rejection of claims 59-64, 66, and 67 under 35 U.S.C. § 112, second paragraph is reversed.
- The Examiner's rejection of claims 64, 66, 67, 72, 74, and 75 under 35 U.S.C. § 314(a) is affirmed.
- The Examiner's rejection of claims 1-5, 17, 20-26, 38-41, 54, 56, 59, 61-62, 68, 70, 71, 76-78, and 80-82 under 35 U.S.C. § 102(b) as anticipated by IBM is reversed.
- The Examiner's rejection of claims 1-7, 11-12, 17, 20-28, 32, 33, 38-44, 48, 49, 54, 56, 59, 61, 62, 68, 70, 71, and 76-82 under 35 U.S.C. § 103(a) as obvious over IBM and Interleaf is affirmed.
- The Examiner's rejection of claims 1-14, 16-18, 20-35, 37-51, 53, 54, 56, 57, 59-62, 64, 66, 68-72, 74, and 76-82 under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, and Interleaf Patent is affirmed.
- The Examiner's rejection of claims 15, 19, 36, and 52 under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and Lieberman is affirmed.

- The Examiner's rejection of claims 1, 2, 4-7, 11, 12, 22-23, 25-28, 32, 33, 38, 39, 41-44, 48, and 49 under 35 U.S.C. § 102(b) as anticipated by Interleaf is affirmed.
- The Examiner's rejection of claims 1, 2, 4-14, 16-18, 22, 23, 25-35, 37-39, 41-51, 53, 54, 57, 61, 68, 70, and 76-82 under 35 U.S.C. § 103(a) as obvious over Interleaf and Interleaf Patent is affirmed.
- The Examiner's rejection of claim 59 under 35 U.S.C. § 103(a) as obvious over Interleaf and Interleaf Patent is reversed.
- The Examiner's rejection of claims 15, 19, 36, 52, and 58 under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and Lieberman is affirmed.
- The Examiner's rejection of claims 1-54, 56-60, 62, 64, 66, 68-71, 76, 77, and 80-82 under 35 U.S.C. § 103(a) as obvious over Interleaf Patent and IBM is affirmed.
- The Examiner's rejection of claim 55 under 35 U.S.C. § 103(a) as obvious over Interleaf, Interleaf Patent, and '705 Patent is reversed.
- The Examiner's rejection of claims 55 and 75 under 35 U.S.C. § 103(a) as obvious over IBM, Interleaf, Interleaf Patent, and '705 Patent is reversed.
- The Examiner's rejection of claims 55, 63, and 67 under 35 U.S.C. § 103(a) as obvious over Interleaf Patent, IBM, and '705 Patent is reversed.

IV. DECISION

The Examiner's decision rejecting claims 1-54, 56-62, 64, 66-72, and 74-82 is affirmed. The Examiner's decision rejecting claims 55 and 63 is reversed.

Requests for extensions of time in this *inter partes* reexamination proceeding are governed by 37 C.F.R. § 1.956. *See* 37 C.F.R. § 41.79.

AFFIRMED-IN-PART

PATENT OWNER:

PEARL COHEN ZEDEK LATZER, LLP
1500 BROADWAY, 12TH FLOOR
NEW YORK, NEW YORK 10036
TEL: (646) 878-0800
FAX: (646) 878-0801

THIRD PARTY REQUESTER:

HAYNES AND BOONE, LLP
IP SECTION, 2323 VICTORY AVENUE, SUITE 700
DALLAS, TEXAS 75219
TELEPHONE: 214/651-5533
FACSIMILE: 214/200-0853
ATTORNEY DOCKET: 38512.24

US006684789B2

(12) **United States Patent**
Krautter

(10) **Patent No.: US 6,684,789 B2**
(45) **Date of Patent: Feb. 3, 2004**

(54) **METHOD AND SYSTEM FOR THE TRANSFORMATION OF DIGITAL PRINT DATA STREAMS AND CORRESPONDING PRINTER AND PRINTER SERVER**

(75) Inventor: **Thomas Erfinders Krautter**, Stuttgart (DE)

(73) Assignee: **CCP Systems AG**, Stuttgart (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/275,784**

(22) PCT Filed: **May 11, 2001**

(86) PCT No.: **PCT/DE01/01796**

§ 371 (c)(1),
(2), (4) Date: **Nov. 7, 2002**

(87) PCT Pub. No.: **WO01/88840**

PCT Pub. Date: **Nov. 22, 2001**

(65) **Prior Publication Data**

US 2003/0140809 A1 Jul. 31, 2003

(30) **Foreign Application Priority Data**

May 17, 2000 (DE) 100 24 177
Jan. 26, 2001 (DE) 101 03 733

(51) **Int. Cl.**[7] **B41F 1/54**
(52) **U.S. Cl.** **101/484**; 101/486; 400/61; 400/62
(58) **Field of Search** 101/484, 485, 101/486; 400/61, 62, 63, 76; 358/1.1, 1.9, 1.15, 1.16, 1.17, 1.18

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,216,754 A 6/1993 Sathi et al.
5,566,278 A * 10/1996 Patel et al. 358/1.15
6,006,013 A 12/1999 Rumph et al.

FOREIGN PATENT DOCUMENTS

EP 0 109 615 B1 6/1994
EP 0 964 339 A2 12/1999
EP 1 061 456 A2 12/2000
GB 2 357 348 A 6/2001
WO WO-00/17748 3/2000
WO WO-00/55720 9/2000

* cited by examiner

Primary Examiner—Andrew H. Hirshfeld
Assistant Examiner—Minh Chau
(74) *Attorney, Agent, or Firm*—Squire, Sanders & Dempsey L.L.P.

(57) **ABSTRACT**

A method for the transformation of digital print data streams, in which an input print data stream (**2**) is read in, this is analyzed by means of a parser (**4**) for graphically representable objects (**5**, **5***a*) and is split up into these graphically representable objects (**5**, **5***a*), and the graphically representable objects (**5**, **5***a*) are stored in a memory (**6**) in an object-oriented format, and the graphically representable objects (**5**, **5***a*) stored in the memory (**6**) in an object-oriented format are transformed into a format for the control of an output device (**9**), preferably a printer, and the objects thus transformed are combined into an output print data stream (**10**) and are output, graphically representable objects (**5**, **5***a*) being stored in the memory (**6**) in an object-oriented format, to which at least one stored script (**5***a*) is assigned, which is executed in the cases defined in the script (**5***a*).

53 Claims, 1 Drawing Sheet

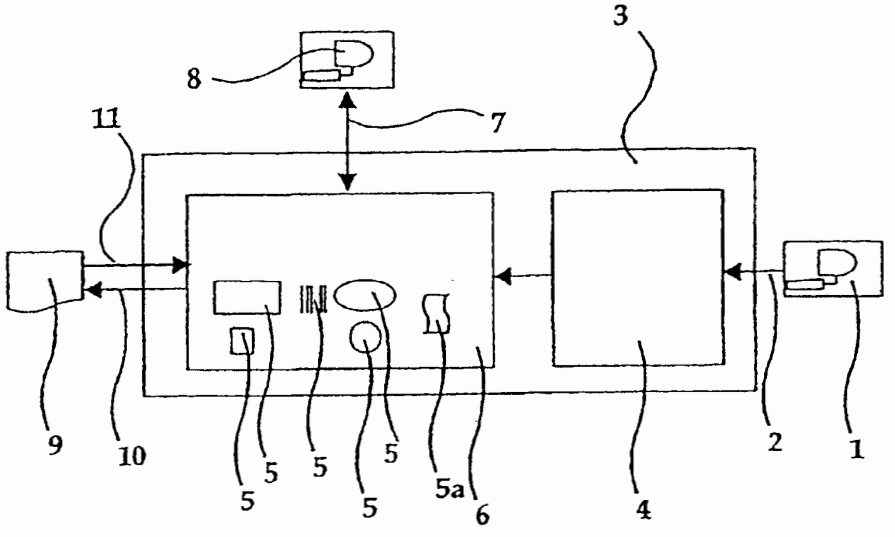
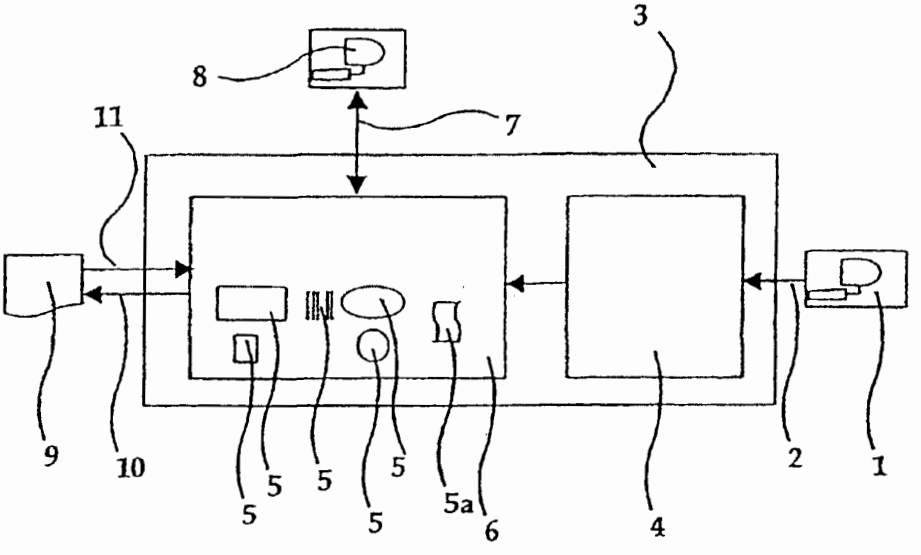


FIG. 1



METHOD AND SYSTEM FOR THE TRANSFORMATION OF DIGITAL PRINT DATA STREAMS AND CORRESPONDING PRINTER AND PRINTER SERVER

The present invention relates to a method and system for the transformation of digital print data streams and corresponding printer and printer server.

Virtually all the output devices which are common nowadays use "page description languages", also called PDL, to produce printed documents. Here, an application program controls a driver for the output device (for example a printer driver). This driver converts information about the graphic objects to be output—for example text or image information—into the respective PDL suitable for the printer used, so that the latter can hereby be controlled directly.

More recent output devices, such as laser printers or digital color printers for example, also offer the possibility of buffering the data streams coming in to control them and, for example, using them as an original form for further incoming print data. This makes it possible to dispense with forms needed for the respective printing, such as letter paper, invoice forms or the like for example, in the individual case. Instead, the application software respectively used merely calls up the form stored once in the printer and combines it with the current print data. In this way, the accumulation of data, for example in networks, can be reduced considerably. However, the result is also organizational advantages: since the forms used no longer have to be kept in reserve by each individual user on his computer, in this way standardized use forms can be achieved, which firstly helps to ensure the often desired standard appearance of a company or an institution and secondly also makes it easier to use current form versions.

However, these aforementioned advantages are normally not used, since the printers used in a company or an institution—with regard to their control—are often not uniform and therefore the use of the functions described above is too complicated, since the appropriate forms either have to be available for each printer model used, which would be very labor-intensive, or only specific printers can be used for specific applications, which is very inflexible.

One possibility of solving this problem is to circumvent the abovedescribed inhomogeneity of the output devices used by employing methods for the conversion of various data stream formats for controlling output devices, which makes it possible for all the computers which produce print data streams to be output to use a standard format for this purpose, by each printer being assigned an interface—be it a dedicated device, be it merely in the form of a software filter—which makes use of such a method and, on the side of the input data stream, uses the format to be used uniformly and, on the side of the output data stream, uses the specific format of the printer to be controlled.

Such a solution is described, for example, by EP 0 109 615 B1, which refers to a method for the conversion of text which is represented in the form of digital data. However, the method taught by this document has considerable disadvantages with regard to the possibilities of current systems from information technology: for example, the method is suitable only for those input print data streams which, in their syntax, follow a format description language whose syntax may be described with the aid of "regular expressions". This is because the method taught in EP 0 109 615 B1 makes use of a status machine, implemented by means of "key status variables", for the recognition and conversion of input control objects recognized in the input print data stream into output control objects. These output control objects are in this case produced directly from the input control objects—specifically in accordance with a fixed assignment—as a function of the respective state representing the key status variables. Such a procedure corresponds to the functioning of the theoretical model of the Moore or Mealy machines, which operate quite efficiently but permit only, the recognition of regular expressions. For these circumstances surrounding information technology at the priority date of EP 0 109 615, such a simple possible transformation may have been sufficient, since—as can already be gathered from claim 1 there—only text had to be converted, apart from format information.

For the current circumstances of PDLs or else other input formats to be recognized where possible, such as HTML or XML, this no longer applies in any way, however. In the meantime, these have been built up in such a complex way with regard to their possibilities that a status machine is no longer in any way adequate for their recognition and conversion.

However, the target format, into which the print data stream is to be transformed, nowadays places high requirements on a transformation: although in principle there would be the possibility here likewise of using the smallest common multiple of the functions of current printing format and in this way of reducing the effort on transformation, this convenience in the design of the transformation process would be brought at great expense in the operation of the method, since in this way the accumulation of data in networks would be increased again, since powerful printer control possibilities which as a rule become more and more specific with regard to the printer type used as the complexity increases, would necessarily have to be dispensed with. Such an increased accumulation of data would, however, again stand in the way of the objective of reducing the data traffic in the network by using PDLs. Thus, at the same time, there is a requirement on the transformation process that the latter produces the preconditions that the target formats can be produced in the most flexible manner possible with all their available printing functions, in order that the traffic on the data transmission lines can thus be minimized.

Furthermore, it is necessary to state that printing systems, even today, still only fulfill a single purpose: namely printing. All the manufacturers of laser printers and digital copying systems have made great efforts in recent years to match the processor powers, storage capacities and additional options (such as memory cards, hard drives, network cards) of these systems to the increasing requirements. However, the manner in which printers and copiers are controlled and programmed has not changed significantly in the last ten years.

Printing systems are still controlled by a page description language (PDL) such as PCL, Postscript or Prescribe. It permits a document and its components to be described adequately. However, the many additional options of modem printing and copying systems available in the meantime cannot be used. The consequence of this is that, even today, the entire printing process is controlled and monitored by a host computer. Its task substantially comprises converting the respective information exactly into the page description language "understood" by the printing system.

It is therefore an object of the present invention to specify a method for the transformation of digital print data streams which is both capable of recognizing more complex page description languages, whose syntax may no longer be described with the aid of simple regular expressions, and also provides the preconditions that the recognized graphic

objects can be transformed into a target format, but also processed further, as flexibly and effectively as possible, that is to say with regard to their description at the highest possible level of abstraction.

According to the invention, this object is achieved by a method for the transformation of digital print data streams, in which an input print data stream is read in, this is analyzed by means of a parser for graphically representable objects and is split up into these graphically representable objects, and the graphically representable objects are stored in a memory in an object-oriented format, and the graphically representable objects stored in the memory in an object-oriented format are transformed into a format for the control of an output device, preferably a printer, and the objects thus transformed are combined into an output print data stream and are output, and which, according to the invention, is characterized in that graphically representable objects are stored in the memory in an object-oriented format, to which at least one stored script is assigned, which is executed in the cases defined in the script.

In this case, as opposed to the use of a status machine, the analysis and splitting of the input print data stream by a parser (syntax analyzer) ensures that the syntax of the page description language is no longer restricted to the use of regular expressions, and thus powerful page description languages can also be used. Instead, such a parser, in terms of its theoretical performance, corresponds to a Turing machine and therefore ensures the theoretically maximum achievable performance for the analysis and splitting up of formal languages.

Furthermore, storing the graphically representable objects—and therefore of course also the scripts, which themselves are certainly also graphically representable objects—in a memory in an object-oriented format achieves the situation where the objects recognized by the parser are then available in this intermediate format which is extremely beneficial for further processing.

The objects are preferably managed here by means of a "display list management", which supports one-page and multi-page documents at as many levels as desired and which can be expanded dynamically by new objects. The individual graphic objects are stored by using their membership of specific—expediently suitably hierarchically organized—classes such as relating to the class of points, ellipses, circles, lines, polygons, rectangles, squares or else to the more complex object types, such as bar codes, more complex texts or freely definable elements such as color profiles or fonts, which permits their effective conversion into an output print data stream, since, through the class of the respective object, there is already implicit information available about its possible transformation into the format of the output print data stream. For example, via an object of the type square, it is already known from the object hierarchy that this is a subclass of the rectangle. If, then, the target format for which an output print data stream is to be produced provides speech constructs relating to the description of rectangles in the page description language, then it is clear, merely on the basis of the position of the square in the object class hierarchy, that this is also a rectangle—albeit with special characteristics—and to this extent the possibilities of the target format with regard to rectangles can also be used for an object in the square class.

In addition to such implicit information—which can be derived from the object class hierarchy—about the individual objects, however, it is also possible to add to the objects explicit information about their possible conversions into specific target formats, it being advantageously possible for this also to be combined with the abovedescribed implicitly provided information, for example by a conversion method into a specific target format being added to a class which is arranged higher in the object class hierarchy, and then automatically also being available to the objects of subordinate, lower-ranking classes by way of inheritance, if a better specified method is not already assigned to said subordinate classes.

In one embodiment of the method according to the invention, the graphically representable objects are combined into super-objects of higher complexity before being stored in the memory.

The super-objects obtained in this way are then stored in the memory in the object-oriented format. In this way, less complex graphic objects can be combined to form more complex graphic super-objects. For example, sequences of lines which in each case join one another at the ends and have been recognized as graphic objects in the input print data stream can be combined to form a graphic polygon super-object. Such a combination offers various advantages, such as easier handling of the super-object stored as a whole as compared with the individual objects, since said super-object can then be treated uniformly by the methods for the super-object class with effect for all the part objects combined in it. It also helps, in certain circumstances, to further minimize the data traffic on the transmission lines used, since an object once combined is subsequently also forwarded in combined form in the output print data stream—if technically supported there—which generally requires a lower data volume to be transmitted than the transmission of the individual objects.

A preferred embodiment of the method according to the present invention is characterized in that a parser is used for the analysis and splitting up into the graphically representable objects, which, in the theoretical model, corresponds to an automatic push-down facility and which is therefore capable of analyzing and splitting up languages with "context-free grammars" particularly effectively.

A further embodiment of the method according to the present invention is characterized in that feedback messages referring to the output print data stream output are read in and are analyzed for error messages which indicate that the output device, preferably the printer, has recognized a transformed graphic object in the output print data stream which cannot be output by said printer, this graphic object is then split up into part objects of lower complexity, and the part objects thus obtained, in the format for the control of the output device, are slipped into the output print data stream which is output to the output device.

In this way, it is likewise possible to test whether the driven output device is, for example, capable of recognizing and outputting a bar-code object directly or not. If it is not capable of this and reports this back, then the bar code is simply split up into objects of the next lower hierarchy, for example filled rectangles, and a further try is made with these objects. This is continued until—if necessary until the graphic objects are split up into individual points—the output attempt is successful. The object-oriented data structure with its object hierarchy, chosen for the intermediate format, also proves to be particularly suitable for this procedure. For the further performance of the method, it is preferably noted at which level of the object classes in each case the splitting process was successful for a specific output device, in order then, in the next attempt, already to begin the output process at this level, in order also thus to avoid unnecessary data transfers, but likewise to utilize the maximum level of abstraction of the output device. In this way,

the data volume to be transmitted is reduced to the necessary extent, even with high flexibility.

In an embodiment of the method according to the present invention, at least one graphically representable object stored in the memory in the object-oriented format is assigned at least one script which controls external devices, preferably archiving devices, folding systems, enveloping systems or security equipment, which permits the incorporation of all the devices needed in the widest sense for document processing.

A further preferred embodiment of the method according to the invention is characterized in that at least one graphically representable object stored in the memory in the object-oriented format is assigned at least one script which automatically receives data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

The script automatically receiving data can preferably also request this data automatically.

It is likewise possible that a script also sends data automatically, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails, it being able in particular also to send the graphic object associated with itself to a receiver.

It can also in turn reassign the data received by it to the graphic object associated with it, and forwards the graphic object associated with itself to a receiver together with the data requested, received and reassigned by itself, or else print out said data.

In relation to the above explanations, it should be noted that the embodiments of the method according to the invention which themselves provide other objects with objects, for example by forwarding them or keeping them ready to receive or for interrogation by a script, for example, are also covered by the term "dynamic object linking" (DOL).

Systems that operate on the method according to the invention, such as printing systems, are capable of sending and receiving e-mails and of printing original print and image data without a printer driver. They are able to store any desired information on hard disks or memory cards and make said data available to all the devices connected in the network and Internet. In other words, they independently undertake demanding tasks in information processing and provision, in order to relieve host computers and personal computers of quite a lot of administrative tasks. In a heterogeneous network and printing environment with laser printing and copying systems from different manufacturers in combination with impact printers and special printing systems, they also make it possible to administer all the connected printing systems with the aid of a single standardized programming language, namely the script language, and therefore reduce the effort on administration to a minimum. At this point, it should be mentioned that these systems operating by the method according to the invention are also designated JScribe (registered trademark) systems and, accordingly, the method according to the invention is also designated JScribe (registered trademark).

When JScribe (registered trademark) is used, developers and system houses will therefore be in a position to provide objects and functions which are stored in resident form in the printing system and permit and control desired individual operating sequences. These objects and functions can use any functionality provided by the JScribe (registered trademark) basic technology, including extremely demanding commands for the job or page processing and for the complete control of the print data and emulations. The method according to the invention preferably also enables access to internal printer functions and status information (page counter, network components, file system and so on), for example via a script.

The method according to the invention is preferably characterized in that graphically representable objects are stored in the memory in an object-oriented format, to which at least one stored script is assigned, which is executed in the case of the output of the object defined in the script. In this way, for example, it is possible to execute such scripts, for example Visual Basic Scripts, Java Scripts or else "stream code" in an event-oriented manner, for example in the case where a form object is printed out, likewise "ON-PRINT" by which means, for example, to execute such functions as the printing of copies of the same form with the same net data but on different paper from different trays. In particular in interaction with those embodiments of the method according to the invention which control external devices, such as folding or enveloping machines or else stapling machines, this is particularly advantageous.

However, it may also be the case that at least one case relating to the execution of the script is defined in the respective script and occurs automatically, preferably without further influence from outside.

For example, the automatically occurring case, defined at least in the respective script, relating to the execution of the script can be configured as a timer, that is to say as a case which occurs automatically as a result of the expiry of a time, this timer preferably operating cyclically, that is to say starting itself again upon expiry.

Automatic scripts can therefore intrinsically become active and, for example, load the daily newspaper, where possible itself assembled from different sources, from the Internet, assign the found, loaded and analyzed information to a stored object and then print this object, completely without the participation of a PC or other host computer to which the printer would be connected.

For example, the simple download of JScribe (registered trademark) sequences (scripts with appropriately associated objects) can, for example, arrange for the printer automatically to fetch information about current share prices, to format it and to print it out. Image information, text documents, web pages, XML documents and any other desired print data can be analyzed while dispensing with any preparation by the PC (for example by a printer driver), modified if necessary and printed out in optimum quality. Since JScribe (registered trademark) can also be employed simultaneously as a server version for computer systems, printing systems are for the first time made capable of accessing stocks of data on host systems (for example SQL databases) interactively during the printing operation.

The language used for the scripts according to the present invention is preferably Java Script. Java Script, as a world-established standard for the script-controlled, intelligent programming of web pages, has triggered in the Internet an avalanche of innovative and functional solutions which have contributed decisively to ringing in the age of eBusiness and eCommerce. This intelligent technology, which has so decisively marked the worldwide, rapid development of the Internet, is therefore now also available for printing systems for the first time and here preferably forms the basic technology for script applications in the area of the present invention, and consequently print and document management, which is certainly uniquely and, as compared with established solutions, considerably more cost-effective.

With JScribe in conjunction with Java Script, an innovative technology is therefore provided which allows any

corresponding print system operated in accordance with the method according to the invention to be programmed just as simply as an Internet homepage. The communications possibilities already described, together with the logically modular object-oriented construction of JScribe and the Java-Script-typical expansion possibilities ideally supporting JScribe permit within the shortest possible time the construction of complex output management systems for an extremely wide range of applications.

A further preferred embodiment of the method according to the present invention is characterized in that the graphically representable objects stored in the memory in an object-oriented format, preferably also script objects (for example Java Script objects), preferably before they are output in the output print data stream, are kept ready by an application interface to be read out, to be changed, to be deleted or to have new objects appended.

According to the prior art, hitherto the page descriptions necessary for the storage of forms in the output devices had to be created laboriously by hand, that is to say programmed in the respective page description language—time-consuming and expensive work which can be carried out only by a few programmers qualified to do this. The same also applies to changes in the stored data.

The object-oriented intermediate format now makes it possible for the stored graphically representable objects to be kept ready to be read out, to be changed, to be deleted or for new objects to be appended, in a technically elegant manner via an application interface, by assigning the methods required for this to the respective objects in accordance with their class hierarchy. This means that the objects stored in the memory can, for example, be displayed on a screen and modified as desired. Here, too, deleting existing objects and appending new objects are also possible.

By means of binding suitable application software—also called FormMaker—it is therefore made possible in particular for each EDP user to modify existing forms and to create new forms entirely without any programming knowledge, which likewise applies to scripts.

Given suitable selection of the application interface and processing methods correspondingly available to a sufficient extent for the object classes used, a graphic core system with a functional interface is thus made available, which can be used by applications for graphic user interfaces, such as those based on the Windows operating system, to display the object data as a standard document on the screen and to modify it with different processing tools.

The application interface also preferably permits script objects, preferably Java Script objects themselves, to be read out graphically, to be changed, to be deleted or to be appended, these graphically performed manipulations being automatically transformed, if required, into script objects, preferably Java Script objects. It thus provides a complete graphic development environment for computers, preferably computers operating under the Windows operating system, which permits the printing and copying systems to be programmed without Java Script knowledge.

In addition, already existing development tools which are based on Java can likewise be used for the development of individual JScribe (registered trademark) applications.

The use of "FormMaker" application software permits the design of "intelligent" electronic forms, which are transformed into logical documents with the aid of JScribe (registered trademark). These in turn can be made available in systems connected to the network and output at any desired location by any desired printing systems, preferably laser printing systems and digital copying systems, sent as e-mail or else transferred to archiving systems.

The present method according to the invention can also be present implemented on a system for the transformation of digital print data streams comprising at least one data processing unit having at least one memory and at least one communications interface, the data processing unit being programmed in such a way that it operates in accordance with an embodiment of the method according to the invention.

In this case, the system preferably also has an operating station with display means and input means, which makes it possible for the graphically representable objects stored in the memory of the data processing unit in an object-oriented format, preferably also script objects, to be read out via the application interface, to be changed, to be deleted or to be appended, preferably before they are output in the output print data stream.

In addition, the system according to the invention can moreover permit respectively stored objects, preferably even script objects themselves, such as Java Script objects, to be read out graphically, to be changed, to be deleted or to be appended, these graphically performed manipulations being transformed automatically, if required, into Java Script objects.

The system according to the invention can also be integrated into a printer or else a printer server.

JScribe (registered trademark) can therefore not only be employed directly on printers and digital copying system but can also be implemented on PC server platforms.

For installation purposes on printing systems, the JScribe script sequences can, for example, be incorporated into a Prescribe (registered trademark) data stream. The printing system receiving this data, for example the appropriate laser printer or digital copier, will read in and compile the program code.

This permits the configuration of networks with hardware units which are small but equipped with high functionality, which have a common interface and permit access relating to archiving documents, to distributed printing (cluster printing) and security printing and much more.

The abovedescribed embodiments of the method according to the present invention can of course in each case also be implemented as a computer program product which has a computer-readable medium with computer program code means or as a computer program on an electronic carrier signal and in which, in each case after the computer program has been loaded, the computer is caused by the program to carry out the method according to the invention described here.

In the following text, an exemplary embodiment, not to be understood as restrictive, will be discussed by using the drawing, in which:

FIG. **1** shows the sequence of an embodiment of the method according to the invention using a schematic representation.

FIG. **1** shows the sequence of an embodiment of the method according to the invention using a schematic representation. From a computer **1**, an input print data stream **2** is sent to a device **3**—for example a computer such as a PC or else an intelligent output device such as an intelligent printer—which operates in accordance with the method according to the present invention. There, the input print data stream **2** is analyzed and split up by a parser **4**. The graphic objects **5**, **5***a* recognized as the product of this splitting are stored in a memory **6** in an object-oriented format; this is after they have possibly been combined to form super-objects. The objects **5** stored in the memory **6**, preferably script objects **5***a*, are kept ready to be read out via

an application interface **7**, to be changed, to be deleted or for new objects to be appended. In this way, the objects **5**, **5***a* stored in the memory **6** can, for example, be displayed on a screen **9** and modified as desired. Deleting existing objects and appending new objects is also possible here. If suitable application software is used, it is thus possible for any user to modify existing forms easily and without programming knowledge or to create new forms easily and without programming knowledge or to create new forms. The graphically representable objects **5**, **5***a* stored in the memory **6** in an object-oriented format are transformed into a format for the control of an output device, preferably a printer **9**, in order to be output, and the objects **5**, **5***a* thus transformed are combined into an output print data stream **10** and output. Feedback messages **11** concerning the output print data stream **10** output are read in and analyzed for error messages which indicate that the printer **10** has detected a graphic object **5**, **5***a* in the output print data stream **10** which cannot be output or processed by said printer. This graphic object **5**, **5***a* is then split up into part objects of lower complexity, and the part objects obtained in this way, in the format for the control of the printer **9**, are slipped into the output print data stream **10** which is output to the printer **9**.

What is claimed is:

1. A method for the transformation of digital print data streams, in which

(i) an input print data stream (**2**) is read in,

(ii) this is analyzed by means of a parser (**4**) for graphically representable objects (**5**) and is split up into these graphically representable objects (**5**), and

(iii) the graphically representable objects (**5**) are stored in a memory (**6**) in an object-oriented format, and

(iv) the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format are transformed into a format for the control of an output device (**9**), preferably a printer, and

(v) the objects thus transformed are combined into an output print data stream (**10**) and are output,

characterized in that graphically representable objects (**5**, **5***a*) are stored in the memory (**6**) in an object-oriented format, to which at least one stored script is assigned, which is executed in the cases defined in the script.

2. The method as claimed in claim **1**, characterized in that the graphically representable objects (**5**, **5***a*) are combined into super-objects of higher complexity before being stored in the memory (**6**).

3. The method as claimed in claim **1**, characterized in that feedback messages (**11**) referring to the output print data stream (**10**) output are read in and are analyzed for error messages which indicate that the output device (**9**), preferably the printer, has recognized a transformed graphic object in the output print data stream (**10**) which cannot be output by said printer, this graphic object is then split up into part objects of lower complexity, and the part objects thus obtained, in the format for the control of the output device (**9**), are slipped into the output print data stream (**10**) which is output to the output device (**9**).

4. The method as claimed in claim **1**, characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which controls external devices, preferably archiving devices, folding systems, enveloping systems or security equipment.

5. The method as claimed in claim **1**, characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically receives data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

6. The method as claimed in claim **5**, characterized in that the script (**5***a*) automatically receiving data also requests this data automatically.

7. The method as claimed in claim **5**, characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with itself, and prints out the graphic object (**5**) assigned to itself together with the data requested, received and reassigned by itself.

8. The method as claimed in claim **1**, characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically sends data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

9. The method as claimed in claim **8**, characterized in that the script (**5***a*) sends the graphic object (**5**) associated with itself to receiver.

10. The method as claimed in claim **9**, characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with it, and prints out the graphic object (**5**) assigned to itself together with the data requested, received and reassigned by itself.

11. The method as claimed in claim **1**, characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which is executed in the case of the output of the object (**5**) defined in the script (**5***a*).

12. The method as claimed in claim **1**, characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*), at least one case relating to the execution of the script (**5***a*) being defined in the respective script (**5***a*), and occurring automatically, preferably without further influence from outside.

13. The method as claimed in claim **12**, characterized in that the automatically occurring case, defined at least in the respective script (**5***a*), relating to the execution of the script (**5***a*) is configured as a timer, that is to say as a case which occurs automatically as a result of expiry of time.

14. The method as claimed in claim **13**, characterized in that the timer operates cyclically, that is to say it starts itself again upon expiry.

15. The method as claimed in claim **1**, characterized in that Java Script is used as a formal language for the scripts.

16. The method as claimed in claim **1**, characterized in that the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format, preferably also script objects (**5***a*), preferably before they are output in the output print data stream (**10**), are kept ready by an application interface (**7**) to be read out, to be changed, to be deleted or to have new objects (**5**) appended.

17. A system for the transformation of digital print data streams comprising at least one data processing unit having at least one memory and at least one communications interface, characterized in that the data processing unit is programmed in such a way that it operates in accordance with the method as claimed claim **1**.

18. The system for the transformation of digital print data streams as claimed in claim **17**, the system also has an operating station with display means (**8**) and input means, which makes it possible for the graphically representable objects (**5**) stored in the memory (**6**) of the data processing

unit in an object-oriented format, preferably also script objects (**5***a*), to be read out via the application interface (**7**), to be changed, to be deleted or to be appended, preferably before they are output in the output print data stream (**10**).

19. The system for the transformation of digital print data streams as claimed in claim **17**, wherein the data processing unit, permits respectively stored objects, preferably also Java Script objects (**5***a*) themselves, to be read out graphically, to be changed, to be deleted or to be appended, these graphically performed manipulations if necessary being transformed automatically into Java Script objects (**5***a*).

20. A printer, characterized in that it has a system for the transformation of digital print data streams as claimed in claim **17**.

21. A printer server, characterized in that it has a system for the transformation of digital print data streams as claimed in claim **17**.

22. A computer-readable medium having stored thereon instructions to cause a processor to execute a method, the method comprising:

(i) an input print data stream (**2**) is read in,

(ii) this is analyzed by means of a parser (**4**) for graphically representable objects (**5**) and is split up into these graphically representable objects (**5**), and

(iii) the graphically representable objects (**5**) are stored in a memory (**6**) in an object-oriented format,

(iv) the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format are transformed into a format for the control of an output device (**9**), preferably a printer, and

(v) the objects thus transformed are combined into an output print data stream (**10**) and are output,

characterized in that graphically representable objects (**5**, **5***a*) are stored in the memory (**6**) in an object-oriented format, to which at least one stored script is assigned, which is executed in the cases defined in the script.

23. The computer-readable medium as claimed in claim **22**, the method characterized in that the graphically representable objects (**5**, **5***a*) are combined into super-objects of higher complexity before being stored in the memory (**6**).

24. The computer-readable medium as claimed in claim **22**, the method characterized in that feedback messages (**11**) referring to the output print data stream (**10**) output are read in and are analyzed for error messages which indicate that the output device (**9**), preferably the printer, has recognized a transformed graphic object in the output print data stream (**10**) which cannot be output by said printer, this graphic object is then split up into part objects of lower complexity, and the part objects thus obtained, in the format for the control of the output device (**9**), are slipped into the output print data stream (**10**) which is output to the output device (**9**).

25. The computer-readable medium as claimed in claim **22**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which controls external devices, preferably archiving devices, folding systems, enveloping systems or security equipment.

26. The computer-readable medium as claimed in claim **22**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically receives data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

27. The computer-readable medium as claimed in claim **26**, the method characterized in that the script (**5***a*) automatically receiving data also requests this data automatically.

28. The computer-readable medium as claimed in claim **26**, the method characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with itself, and prints out the graphic object (**5**) assigned to itself together with the data requested, received and reassigend by itself.

29. The computer-readable medium as claimed in claim **22**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically sends data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

30. The computer-readable medium as claimed in claim **29**, the method characterized in that the script (**5***a*) sends the graphic object (**5**) associated with itself to a receiver.

31. The computer-readable medium as claimed in claim **30**, the method characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with it, and forwards the graphic object (**5**) associated with itself to a receiver together with the data requested, received and reassigned by itself.

32. The computer-readable medium as claimed in claim **22**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which is executed in the case of the output of the object (**5**) defined in the script (**5***a*).

33. The computer-readable medium as claimed in claim **22**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*), at least one case relating to the execution of the script (**5***a*) being defined in the respective script (**5***a*), and occurring automatically, preferably without further influence from outside.

34. The computer-readable medium as claimed in claim **33**, the method characterized in that the automatically occurring case, defined at least in the respective script (**5***a*), relating to the execution of the script (**5***a*) is configured as a timer, that is to say as a case which occurs automatically as a result of expiry of time.

35. The computer-readable medium as claimed in claim **34**, the method characterized in that the timer operates cyclically, that is to say it starts itself again upon expiry.

36. The computer-readable medium as claimed in claim **22**, the method characterized in that Java Script is used as a formal language for the scripts.

37. The computer-readable medium as claimed in claim **22**, the method characterized in that the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format, preferably also script objects (**5***a*), preferably before they are output in the output print data stream (**10**), are kept ready by an application interface (**7**) to be read out, to be changed, to be deleted or to have new objects (**5**) appended.

38. A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by a processor, cause the processor to perform a method, the method comprising:

(i) an input print data stream (**2**) is read in,

(ii) this is analyzed by means of a parser (**4**) for graphically representable objects (**5**) and is split up into these graphically representable objects (**5**), and

(iii) the graphically representable objects (**5**) are stored in a memory (**6**) in an object-oriented format,

(iv) the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format are transformed into a format for the control of an output device (**9**), preferably a printer, and

(v) the objects thus transformed are combined into an output print data stream (**10**) and are output,

characterized in that graphically representable objects (**5**, **5***a*) are stored in the memory (**6**) in an object-oriented format, to which at least one stored script is assigned, which is executed in the cases defined in the script.

39. The computer data signal as claimed in claim **38**, the method characterized in that the graphically representable objects (**5**, **5***a*) are combined into super-objects of higher complexity before being stored in the memory (**6**).

40. The computer data signal as claimed in claim **38**, the method characterized in that feedback messages (**11**) referring to the output print data stream (**10**) output are read in and are analyzed for error messages which indicate that the output device (**9**), preferably the printer, has recognized a transformed graphic object in the output print data stream (**10**) which cannot be output by said printer, this graphic object is then split up into part objects of lower complexity, and the part objects thus obtained, in the format for the control of the output device (**9**), are slipped into the output print data stream (**10**) which is output to the output device (**9**).

41. The computer data signal as claimed in claim **38**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which controls external devices, preferably archiving devices, folding systems, enveloping systems or security equipment.

42. The computer data signal as claimed in claim **38**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically receives data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

43. The computer data signal as claimed in claim **42**, the method characterized in that the script (**5***a*) automatically receiving data also requests this data automatically.

44. The computer data signal as claimed in claim **42**, the method characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with itself, and prints out the graphic object (**5**) assigned to itself together with the data requested, received and reassigned by itself.

45. The computer data signal as claimed in claim **38**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which automatically sends data, preferably data organized in an object-oriented manner, image data, text data or data from web pages from the Internet, data from XML documents or else e-mails.

46. The computer data signal as claimed in claim **45**, the method characterized in that the script (**5***a*) sends the graphic object (**5**) associated with itself to a receiver.

47. The computer data signal as claimed in claim **46**, the method characterized in that the script (**5***a*) in turn reassigns the data received by it to the graphic object (**5**) associated with it, and forwards the graphic object (**5**) associated with itself to a receiver together with the data requested, received and reassigned by itself.

48. The computer data signal as claimed in claim **38**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*) which is executed in the case of the output of the object (**5**) defined in the script (**5***a*).

49. The computer data signal as claimed in claim **38**, the method characterized in that at least one graphically representable object (**5**) stored in the memory (**6**) in the object-oriented format is assigned at least one script (**5***a*), at least one case relating to the execution of the script (**5***a*) being defined in the respective script (**5***a*), and occurring automatically, preferably without further influence from outside.

50. The computer data signal as claimed in claim **49**, the method characterized in that the automatically occurring case, defined at least in the respective script (**5***a*), relating to the execution of the script (**5***a*) is configured as a timer, that is to say as a case which occurs automatically as a result of expiry of time.

51. The computer data signal as claimed in claim **50**, the method characterized in that the timer operates cyclically, that is to say it starts itself again upon expiry.

52. The computer data signal as claimed in claim **38**, the method characterized in that Java Script is used as a formal language for the scripts.

53. The computer data signal as claimed in claim **38**, the method characterized in that the graphically representable objects (**5**) stored in the memory (**6**) in an object-oriented format, preferably also script objects (**5***a*), preferably before they are output in the output print data stream (**10**), are kept ready by an application interface (**7**) to be read out, to be changed, to be deleted or to have new objects (**5**) appended.

* * * * *